



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° 03**

**NOMBRE COMPLETO:** Medrano Miranda Daniel Ulises

**N° de Cuenta:** 318045351

**GRUPO DE LABORATORIO:** 03

**GRUPO DE TEORÍA:** 04

**SEMESTRE 2025-1**

**FECHA DE ENTREGA LÍMITE:** 27/Agosto/2024

**CALIFICACIÓN:** \_\_\_\_\_

## EJERCICIOS DE SESIÓN:

### 1. Actividades realizadas. Una descripción de los ejercicios y capturas de pantalla de bloques de código generados y de ejecución del programa

1.- Instanciar primitivas geométricas para recrear el dibujo de la práctica pasada en 3D, se requiere que exista un piso; la casa tiene una ventana azul circular justo en medio de la pared trasera, 2 ventanas verdes en cada pared lateral iguales a las de la pared frontal y solo puerta en la pared frontal.

```
387 // - - - EJERCICIO DE CLASE - - -
388 // CUBO ROJO PAREDES CASA
389 model = glm::mat4(1.0f);
390 //Traslación inicial para posicionar en -Z a los objetos
391 model = glm::translate(model, glm::vec3(0.0f, 0.0f, -5.0f));
392 //otras transformaciones para el objeto
393 model = glm::scale(model, glm::vec3(1.5f, 1.5f, 1.5f));
394 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
395 //La línea de proyección solo se manda una vez a menos que en tiempo de ejecución
396 //se programe cambio entre proyección ortogonal y perspectiva
397 glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
398 glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
399 color = glm::vec3(1.0f, 0.0f, 0.0f);
400 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
401 meshList[0]-->RenderMesh(); //dibuja cubo y pirámide triangular
402 //meshList[3]-->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
403 //sp.render(); //dibuja esfera
404
405 //VENTANA VERDE ENFRETE IZQUIERDA
406 model = glm::mat4(1.0f);
407 color = glm::vec3(0.0f, 1.0f, 0.0f);
408 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
409 model = glm::translate(model, glm::vec3(-0.3f, 0.3f, -4.44f));
410 model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.4f));
411 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
412 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
413 meshList[0]-->RenderMesh(); //dibuja cubo y pirámide triangular
414 //meshList[3]-->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
415 //sp.render(); //dibuja esfera
416
417 //VENTANA VERDE ENFRETE DERECHA
418 model = glm::mat4(1.0f);
419 //color = glm::vec3(0.0f, 1.0f, 0.0f);
420 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
421 model = glm::translate(model, glm::vec3(0.3f, 0.3f, -4.44f));
422 model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.4f));
423 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
424 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
425 meshList[0]-->RenderMesh(); //dibuja cubo y pirámide triangular
426 //meshList[3]-->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
427 //sp.render(); //dibuja esfera
428
429 //PUERTA VERDE ENFRETE
430 model = glm::mat4(1.0f);
431 //color = glm::vec3(0.0f, 1.0f, 0.0f);
432 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
433 model = glm::translate(model, glm::vec3(0.0f, -0.546f, -4.44f));
434 model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.4f));
435 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
436 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
437 meshList[0]-->RenderMesh(); //dibuja cubo y pirámide triangular
438 //meshList[3]-->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
439 //sp.render(); //dibuja esfera
440
```

```

441 //VENTANA VERDE DERECHA LADO IZQUIERDO
442 model = glm::mat4(1.0f);
443 //color = glm::vec3(0.0f, 1.0f, 0.0f);
444 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
445 model = glm::translate(model, glm::vec3(-0.56f, 0.3f, -4.7f));
446 model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.4f));
447 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
448 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
449 meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
450 //meshList[3]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
451 //sp.render(); //dibuja esfera
452
453 //VENTANA VERDE IZQUIERDA LADO IZQUIERDO
454 model = glm::mat4(1.0f);
455 //color = glm::vec3(0.0f, 1.0f, 0.0f);
456 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
457 model = glm::translate(model, glm::vec3(-0.56f, 0.3f, -5.3f));
458 model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.4f));
459 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
460 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
461 meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
462 //meshList[3]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
463 //sp.render(); //dibuja esfera
464
465 //VENTANA VERDE IZQUIERDA LADO DERECHO
466 model = glm::mat4(1.0f);
467 //color = glm::vec3(0.0f, 1.0f, 0.0f);
468 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
469 model = glm::translate(model, glm::vec3(0.56f, 0.3f, -4.7f));
470 model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.4f));
471 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
472 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
473 meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
474 //meshList[3]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
475 //sp.render(); //dibuja esfera
476
477 //VENTANA VERDE DERECHA LADO DERECHO
478 model = glm::mat4(1.0f);
479 //color = glm::vec3(0.0f, 1.0f, 0.0f);
480 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
481 model = glm::translate(model, glm::vec3(0.56f, 0.3f, -5.3f));
482 model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.4f));
483 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
484 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
485 meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
486 //meshList[3]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
487 //sp.render(); //dibuja esfera
488
489 //TECHO AZUL ARRIBA
490 model = glm::mat4(1.0f);
491 color = glm::vec3(0.0f, 0.0f, 1.0f);
492 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
493 model = glm::translate(model, glm::vec3(0.0f, 0.0f, 1.001f, -5.0f));
494 model = glm::scale(model, glm::vec3(1.5f, 0.5f, 1.5f));
495 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
496 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
497 //meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
498 meshList[4]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
499 //sp.render(); //dibuja esfera
500
501 //VENTANA CILINDRICA AZUL ATRÁS
502 model = glm::mat4(1.0f);
503 //color = glm::vec3(0.0f, 0.0f, 1.0f);
504 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
505 model = glm::translate(model, glm::vec3(0.0f, 0.0f, -5.51f));
506 model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.5f));
507 model = glm::rotate(model, 90 * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
508 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
509 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
510 //meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
511 meshList[2]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
512 //sp.render(); //dibuja esfera
513

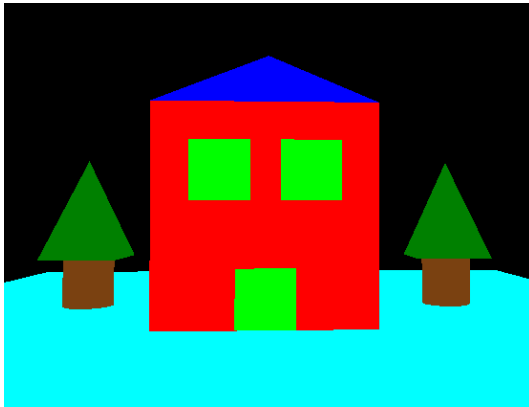
```

```

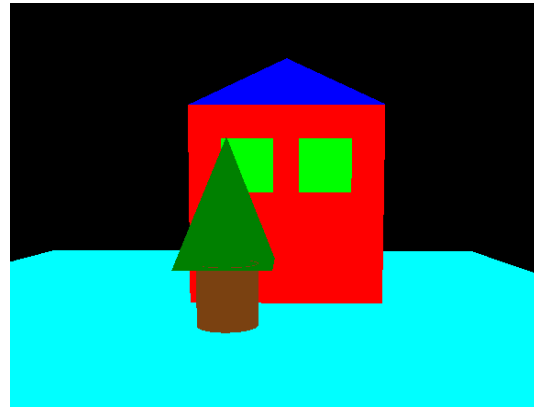
514 //TRONCO CAFE CILINDRICO IZQUIERDA
515 model = glm::mat4(1.0f);
516 color = glm::vec3(0.478f, 0.255f, 0.067f);
517 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
518 model = glm::translate(model, glm::vec3(-1.5f, -0.546f, -5.375f));
519 model = glm::scale(model, glm::vec3(0.2f, 0.4f, 0.2f));
520 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
521 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
522 //meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
523 meshList[2]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
524 //sp.render(); //dibuja esfera
525
526 //TRONCO CAFE CILINDRICO DERECHA
527 model = glm::mat4(1.0f);
528 //color = glm::vec3(0.478f, 0.255f, 0.067f);
529 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
530 model = glm::translate(model, glm::vec3(1.5f, -0.546f, -5.375f));
531 model = glm::scale(model, glm::vec3(0.2f, 0.4f, 0.2f));
532 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
533 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
534 //meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
535 meshList[2]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
536 //sp.render(); //dibuja esfera
537
538 //ÁRBOL VERDE OSCURO PIRAMIDE CUADRANGULAR IZQUIERDA
539 model = glm::mat4(1.0f);
540 color = glm::vec3(0.0f, 0.5f, 0.0f);
541 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
542 model = glm::translate(model, glm::vec3(-1.5f, 0.05f, -5.375f));
543 model = glm::scale(model, glm::vec3(0.6f, 0.8f, 0.6f));
544 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
545 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
546 //meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
547 meshList[4]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
548 //sp.render(); //dibuja esfera
549
550 //ÁRBOL VERDE OSCURO PIRAMIDE CUADRANGULAR DERECHA
551 model = glm::mat4(1.0f);
552 //color = glm::vec3(0.0f, 0.5f, 0.0f);
553 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
554 model = glm::translate(model, glm::vec3(1.5f, 0.05f, -5.375f));
555 model = glm::scale(model, glm::vec3(0.6f, 0.8f, 0.6f));
556 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
557 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
558 //meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
559 meshList[4]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
560 //sp.render(); //dibuja esfera
561
562 //PISO DEL ÁREA CUBO CYAN
563 model = glm::mat4(1.0f);
564 color = glm::vec3(0.0f, 1.0f, 1.0f);
565 //Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
566 model = glm::translate(model, glm::vec3(0.0f, -0.8f, -5.375f));
567 model = glm::scale(model, glm::vec3(6.0f, 0.1f, 6.0f));
568 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
569 glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
570 meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
571 //meshList[4]->RenderMeshGeometry(); //dibuja las figuras geométricas cilindro, cono, pirámide base cuadrangular
572 //sp.render(); //dibuja esfera
573

```

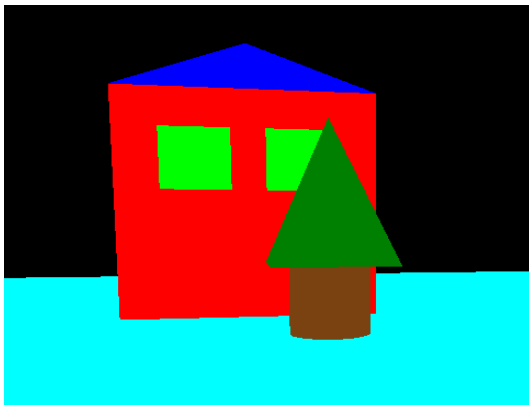
Para la elaboración de esta actividad se copió el código base y modificó colores, figuras y cada una se trasladó, escaló y en un solo caso rotó para poder formar la figura 3D que queda de la siguiente manera:



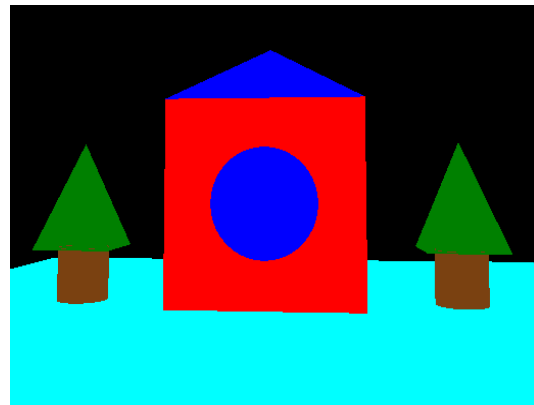
Vista Frontal



Vista Izquierda



Vista Derecha



Vista Trasera

## ***2. Problemas presentados. Listar si surgieron problemas a la hora de ejecutar el código***

A la hora de ejecutar el código no surgió ningún problema, lo más complejo se trató de realizar las traslaciones y escalamientos correctamente, en esta ocasión se tuvo que hacer “a prueba y error” para poder encontrar los valores en los que los ejes X, Y e Z se tenían que mover, sin embargo, considero que con más práctica se puede realizar de manera más fácil.

## ***3. Conclusión:***

### ***a. Los ejercicios de la clase: Complejidad, explicación***

En esta ocasión no sentí mucha complejidad en la elaboración del ejercicio, sin embargo, si tuve que realizar varias modificaciones a los valores de las escalas y traslaciones para que las figuras quedaran exactamente donde yo quería.

***b. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias.***

En esta ocasión no tengo comentarios extras, sólo que me gustó que ahora podemos véer la cámara y trasladarnos en el mundo, sólo debo acostumbrarme a la misma.