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 Nº 03

NOMBRE COMPLETO: García Soto Jean Carlo

Nº de Cuenta: 319226304

GRUPO DE LABORATORIO: 03

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 01/03/2025

CALIFICACIÓN: _____

REPORTE DE PRÁCTICA:

1.- Actividades:

- Generar una pirámide Rubik (pyraminx) de 9 pirámides por cara. Cada cara de la pyraminx que se vea de un color diferente y que se vean las separaciones entre instancias (las líneas oscuras son las que permiten diferencias cada pirámide pequeña).

```
25
26      //Dimensiones de la ventana
27      const float toRadians = 3.14159265f / 180.0; //grados a radianes
28      const float PI = 3.14159265f;
29      float angulo = 1.0f;
30
```

Se declara una variable flotante ***angulo*** esto para poder rotar nuestra figura final y que tenga un angulo tipo variable.

```
96      };
97      GLfloat vertices_piramide_triangular[] = {
98          /*-0.5f, -0.5f,0.0f,     //0
99          0.5f,-0.5f,0.0f,     //1
100         0.0f,0.5f, -0.25f,   //2
101         0.0f,-0.5f,-0.5f,    //3
102         */
103         //Piramide triangular equilatero
104         0.0f, 0.0f, 0.0f,        //0
105         -0.5f, 0.0f, -0.866f,    //1
106         0.5f, 0.0f, -0.866f,     //2
107         0.0f,0.816f,-0.577f      //3
108
109     };
```

Primero se modificaron los vertices de la piramide, esto con el motivo de que la piramide sea de triangulos equilateros, ya que el anterior no cumple con esas especificaciones.

```
365         //-------------- Intancias de Piramides
366         //Piramide principal (Negra)
367         model = glm::mat4(1.0f);
368         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
369         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
370         model = glm::translate(model, glm::vec3(0.0f, -2.0f, -5.0f));
371         model = glm::scale(model, glm::vec3(11.0f, 11.0f, 11.0f));
372         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
373         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
374         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
375         color = glm::vec3(0.0f, 0.0f, 0.0f);
376         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
377         meshList[1]->RenderMesh();
```

Se crea una pirámide negra, en la cual vamos a estar trabajando, y andar pegando o juntando otras pirámides y hacer que se parezca al Pyraminx.

Aquí empezamos creando cada pirámide, primero se empezó por la cara verde, el cual se fue marcando con comentarios cual representaba cada modelo. A cada piramide creada se les hace un translate a la coordenada (0,0,0) esto con el motivo para que cuando se roten todas las figuras, rotan sobre el mismo punto de referencia y no se descuadren. Ya después se les volvió a trasladar al lugar donde pertenecen.

```
//---------------------------------- Inicio piramides verdes ----------------------------------
//Abajo derecha
model = glm::mat4(1.0f);
model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
model = glm::translate(model, glm::vec3(-0.3f, -1.75f, -5.6f));
model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
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));
meshList[1]->RenderMesh();

//Derecha abajo invertido
//Abajo derecha
model = glm::mat4(1.0f);
model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
model = glm::translate(model, glm::vec3(-1.9f, -1.75f, -8.35f));
model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
model = glm::rotate(model, glm::radians(60.0f), glm::vec3(1.0f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(60.0f), glm::vec3(0.0f, 0.0f, 1.0f));
model = glm::rotate(model, glm::radians(320.0f), glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 0.0f, 1.0f));
model = glm::rotate(model, glm::radians(346.0f), glm::vec3(0.0f, 1.0f, 0.0f));
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));
meshList[1]->RenderMesh();
```

```
//Abajo en medio
model = glm::mat4(1.0f);
model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
model = glm::translate(model, glm::vec3(-1.95f, -1.75f, -8.5f));
model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
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));
meshList[1]->RenderMesh();

//Abajo izquierda
model = glm::mat4(1.0f);
model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
model = glm::translate(model, glm::vec3(-3.55f, -1.75f, -11.3f));
model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
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));
meshList[1]->RenderMesh();
```

```
438         //Derecha abajo invertido
439         //Abajo derecha
440         model = glm::mat4(1.0f);
441         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
442         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
443         model = glm::translate(model, glm::vec3(-3.5f, -1.75f, -11.2f));
444         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
445         model = glm::rotate(model, glm::radians(60.0f), glm::vec3(1.0f, 0.0f, 0.0f));
446         model = glm::rotate(model, glm::radians(60.0f), glm::vec3(0.0f, 0.0f, 1.0f));
447         model = glm::rotate(model, glm::radians(320.0f), glm::vec3(0.0f, 1.0f, 0.0f));
448         model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 0.0f, 1.0f));
449         model = glm::rotate(model, glm::radians(346.0f), glm::vec3(0.0f, 1.0f, 0.0f));
450         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
451         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
452         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
453         color = glm::vec3(0.0f, 1.0f, 0.0f);
454         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
455         meshList[1]->RenderMesh();
456
457         //En medio derecha
458         model = glm::mat4(1.0f);
459         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
460         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
461         model = glm::translate(model, glm::vec3(-0.3f, 1.0f, -7.5f));
462         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
463         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
464         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
465         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
466         color = glm::vec3(0.0f, 1.0f, 0.0f);
467         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
468         meshList[1]->RenderMesh();
469
470         //En medioinvertido
471         model = glm::mat4(1.0f);
472         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
473         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
474         model = glm::translate(model, glm::vec3(-1.8f, 1.2f, -10.2f));
475         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
476         model = glm::rotate(model, glm::radians(60.0f), glm::vec3(1.0f, 0.0f, 0.0f));
477         model = glm::rotate(model, glm::radians(60.0f), glm::vec3(0.0f, 0.0f, 1.0f));
478         model = glm::rotate(model, glm::radians(320.0f), glm::vec3(0.0f, 1.0f, 0.0f));
479         model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 0.0f, 1.0f));
480         model = glm::rotate(model, glm::radians(346.0f), glm::vec3(0.0f, 1.0f, 0.0f));
481         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
482         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
483         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
484         color = glm::vec3(0.0f, 1.0f, 0.0f);
485         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
486         meshList[1]->RenderMesh();
487
488         //En medio izquierda
489         model = glm::mat4(1.0f);
490         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
491         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
492         model = glm::translate(model, glm::vec3(-1.9f, 1.0f, -10.3f));
493         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
494         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
495         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
496         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
497         color = glm::vec3(0.0f, 1.0f, 0.0f);
498         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
499         meshList[1]->RenderMesh();
500
```

Aquí acaban las pirámides verdes y se empiezan a crear las rojas.

```
//Arriba
model = glm::mat4(1.0f);
model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
model = glm::translate(model, glm::vec3(-0.25f, 3.75f, -9.4f));
model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
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));
meshList[1]->RenderMesh();
/**/
//--------------------------    Fin de priamides verdes         ----------------------------

//---------------------------------- Inicio piramides rojas ----------------------------------

//Arriba
model = glm::mat4(1.0f);
model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
model = glm::translate(model, glm::vec3(0.25f, 3.75f, -9.45f));
model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
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(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
meshList[1]->RenderMesh();
```

```
//Abajo izquierda
model = glm::mat4(1.0f);
model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
model = glm::translate(model, glm::vec3(0.25f, -1.75f, -5.6f));
model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
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(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
meshList[1]->RenderMesh();

//Abajo izquierda invertida
model = glm::mat4(1.0f);
model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
model = glm::translate(model, glm::vec3(2.0f, -1.75f, -8.30f));
model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
model = glm::rotate(model, glm::radians(70.0f), glm::vec3(1.0f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(10.0                    0.0f, 0.0f, 1.0f));
model = glm::rotate(model, glm::radians(25.0                    0.0f, 1.0f, 0.0f));
model = glm::rotate(model, glm::radians(-15.                    (1.0f, 0.0f, 0.0f));
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(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
meshList[1]->RenderMesh();
```

```
560
561         //Abajo en medio
562         model = glm::mat4(1.0f);
563         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
564         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
565         model = glm::translate(model, glm::vec3(1.95f, -1.75f, -8.5f));
566         //model = glm::rotate(model, glm::radians(60.0f), glm::vec3(0.0f, 1.0f, 0.0f));
567         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
568         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
569         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
570         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
571         color = glm::vec3(1.0f, 0.0f, 0.0f);
572         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
573         meshList[1]->RenderMesh();
574
575         //Abajo derecha invertida
576         model = glm::mat4(1.0f);
577         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
578         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
579         model = glm::translate(model, glm::vec3(3.45f, -1.75f, -11.2f));
580         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
581         model = glm::rotate(model, glm::radians(70.0f), glm::vec3(1.0f, 0.0f, 0.0f));
582         model = glm::rotate(model, glm::radians(10.0f), glm::vec3(0.0f, 0.0f, 1.0f));
583         model = glm::rotate(model, glm::radians(25.0f), glm::vec3(0.0f, 1.0f, 0.0f));
584         model = glm::rotate(model, glm::radians(-15.0f), glm::vec3(1.0f, 0.0f, 0.0f));
585         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
586         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
587         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
588         color = glm::vec3(1.0f, 0.0f, 0.0f);
589         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
590         meshList[1]->RenderMesh();
```

```
592
593             //Abajo derecha
594             model = glm::mat4(1.0f);
595             model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
596             model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
597             model = glm::translate(model, glm::vec3(3.55f, -1.75f, -11.3f));
598             model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
599             glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
600             glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
601             glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
602             color = glm::vec3(1.0f, 0.0f, 0.0f);
603             glUniform3fv(uniformColor, 1, glm::value_ptr(color));
604             meshList[1]->RenderMesh();
605
606             //En medio izquierda
607             model = glm::mat4(1.0f);
608             model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
609             model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
610             model = glm::translate(model, glm::vec3(0.3f, 1.0f, -7.5f));
611             model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
612             glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
613             glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
614             glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
615             color = glm::vec3(1.0f, 0.0f, 0.0f);
616             glUniform3fv(uniformColor, 1, glm::value_ptr(color));
617             meshList[1]->RenderMesh();
```

```
618
619             //En medio invertida
620             model = glm::mat4(1.0f);
621             model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
622             model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
623             model = glm::translate(model, glm::vec3(1.7f, 1.1f, -10.21f));
624             model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
625             model = glm::rotate(model, glm::radians(70.0f), glm::vec3(1.0f, 0.0f, 0.0f));
626             model = glm::rotate(model, glm::radians(10.0f), glm::vec3(0.0f, 0.0f, 1.0f));
627             model = glm::rotate(model, glm::radians(25.0f), glm::vec3(0.0f, 1.0f, 0.0f));
628             model = glm::rotate(model, glm::radians(-15.0f), glm::vec3(1.0f, 0.0f, 0.0f));
629             glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
630             glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
631             glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
632             color = glm::vec3(1.0f, 0.0f, 0.0f);
633             glUniform3fv(uniformColor, 1, glm::value_ptr(color));
634             meshList[1]->RenderMesh();
635
636             //En medio derecha
637             model = glm::mat4(1.0f);
638             model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
639             model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
640             model = glm::translate(model, glm::vec3(1.9f, 1.0f, -10.3f));
641             model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
642             glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
643             glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
644             glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
645             color = glm::vec3(1.0f, 0.0f, 0.0f);
646             glUniform3fv(uniformColor, 1, glm::value_ptr(color));
647             meshList[1]->RenderMesh();
648
649             //----------------------------    Fin de priamides rojas ----------------------------
650
```

(float)(0.0F)

Aquí empiezan a crearse las pirámides azules.

```
650
651         //---------------------------------- Inicio piramides azul ----------------------------------
652
653         //Abajo izquierda
654         model = glm::mat4(1.0f);
655         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
656         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
657         model = glm::translate(model, glm::vec3(3.2f, -1.75f, -11.85f));
658         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
659         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
660         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
661         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
662         color = glm::vec3(0.0f, 0.0f, 1.0f);
663         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
664         meshList[1]->RenderMesh();
665
666         //Abajo izquierda invertido
667         model = glm::mat4(1.0f);
668         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
669         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
670         model = glm::translate(model, glm::vec3(1.63f, -1.6f, -14.4f));
671         model = glm::rotate(model, glm::radians(109.0f), glm::vec3(1.0f, 0.0f, 0.0f));
672         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
673         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
674         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
675         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
676         color = glm::vec3(0.0f, 0.0f, 1.0f);
677         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
678         meshList[1]->RenderMesh();
```

```
Practica3                                   (Ámbito global)
680         //Abajo en medio
681         model = glm::mat4(1.0f);
682         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
683         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
684         model = glm::translate(model, glm::vec3(0.0f, -1.75f, -11.85f));
685         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
686         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
687         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
688         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
689         color = glm::vec3(0.0f, 0.0f, 1.0f);
690         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
691         meshList[1]->RenderMesh();
692
693         //Abajo derecha
694         model = glm::mat4(1.0f);
695         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
696         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
697         model = glm::translate(model, glm::vec3(-3.2f, -1.75f, -11.85f));
698         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
699         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
700         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
701         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
702         color = glm::vec3(0.0f, 0.0f, 1.0f);
703         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
704         meshList[1]->RenderMesh();
705
```

```
706         //Abajo izquierda invertido
707         model = glm::mat4(1.0f);
708         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
709         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
710         model = glm::translate(model, glm::vec3(-1.63f, -1.6f, -14.4f));
711         model = glm::rotate(model, glm::radians(109.0f), glm::vec3(1.0f, 0.0f, 0.0f));
712         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
713         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
714         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
715         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
716         color = glm::vec3(0.0f, 0.0f, 1.0f);
717         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
718         meshList[1]->RenderMesh();
719
720         //En medio izquierda
721         model = glm::mat4(1.0f);
722         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
723         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
724         model = glm::translate(model, glm::vec3(1.65f, 1.0f, -10.9f));
725         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
726         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
727         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
728         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
729         color = glm::vec3(0.0f, 0.0f, 1.0f);
730         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
731         meshList[1]->RenderMesh();
```

```
Practica3                                                    (Ámbito global)
733         //En medio invertido
734         model = glm::mat4(1.0f);
735         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
736         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
737         model = glm::translate(model, glm::vec3(0.0f, 1.1f, -13.45f));
738         model = glm::rotate(model, glm::radians(109.0f), glm::vec3(1.0f, 0.0f, 0.0f));
739         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
740         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
741         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
742         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
743         color = glm::vec3(0.0f, 0.0f, 1.0f);
744         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
745         meshList[1]->RenderMesh();
746
747
748         //En medio derecha
749         model = glm::mat4(1.0f);
750         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
751         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
752         model = glm::translate(model, glm::vec3(-1.6f, 1.0f, -10.9f));
753         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
754         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
755         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
756         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
757         color = glm::vec3(0.0f, 0.0f, 1.0f);
758         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
759         meshList[1]->RenderMesh();
```

Terminan de hacerse las azules y empiezan a crearse las amarillas.

```
761         //Arriba
762         model = glm::mat4(1.0f);
763         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
764         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
765         model = glm::translate(model, glm::vec3(0.0f, 3.75f, -9.9f));
766         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
767         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
768         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
769         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
770         color = glm::vec3(0.0f, 0.0f, 1.0f);
771         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
772         meshList[1]->RenderMesh();
773
774         //-------------------------    Fin de priamides azules         ------------------------------
775
776         //--------------------------------- Inicio piramides amarillas ------------------------------
777
778         //arriba
779         model = glm::mat4(1.0f);
780         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
781         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
782         model = glm::translate(model, glm::vec3(0.0f, -2.01f, -5.8f));
783         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
784         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
785         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
786         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
787         color = glm::vec3(1.0f, 1.0f, 0.0f);
788         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
789         meshList[1]->RenderMesh();
790
```

```
791         //Abajo en medio
792         model = glm::mat4(1.0f);
793         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
794         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
795         model = glm::translate(model, glm::vec3(0.0f, -2.01f, -11.6f));
796         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
797         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
798         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
799         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
800         color = glm::vec3(1.0f, 1.0f, 0.0f);
801         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
802         meshList[1]->RenderMesh();
803
804         //Abajo derecha
805         model = glm::mat4(1.0f);
806         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
807         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
808         model = glm::translate(model, glm::vec3(3.3f, -2.01f, -11.6f));
809         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
810         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
811         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
812         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
813         color = glm::vec3(1.0f, 1.0f, 0.0f);
814         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
815         meshList[1]->RenderMesh();
```

```
817         //Abajo derecha invertido
818         model = glm::mat4(1.0f);
819         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
820         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
821         model = glm::translate(model, glm::vec3(1.65f, -2.01f, -14.0f));
822         model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 1.0f, 0.0f));
823         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
824         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
825         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
826         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
827         color = glm::vec3(1.0f, 1.0f, 0.0f);
828         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
829         meshList[1]->RenderMesh();
830
831         //Abajo izquierda
832         model = glm::mat4(1.0f);
833         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
834         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
835         model = glm::translate(model, glm::vec3(-3.3f, -2.01f, -11.6f));
836         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
837         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
838         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
839         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
840         color = glm::vec3(1.0f, 1.0f, 0.0f);
841         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
842         meshList[1]->RenderMesh();
```

```
844         //Abajo izquierda invertido
845         model = glm::mat4(1.0f);
846         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
847         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
848         model = glm::translate(model, glm::vec3(-1.65f, -2.01f, -14.0f));
849         model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 1.0f, 0.0f));
850         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
851         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
852         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
853         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
854         color = glm::vec3(1.0f, 1.0f, 0.0f);
855         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
856         meshList[1]->RenderMesh();
857
858         //En medio izquierda
859         model = glm::mat4(1.0f);
860         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
861         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
862         model = glm::translate(model, glm::vec3(-1.6f, -2.01f, -8.6f));
863         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
864         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
865         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
866         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
867         color = glm::vec3(1.0f, 1.0f, 0.0f);
868         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
869         meshList[1]->RenderMesh();
```

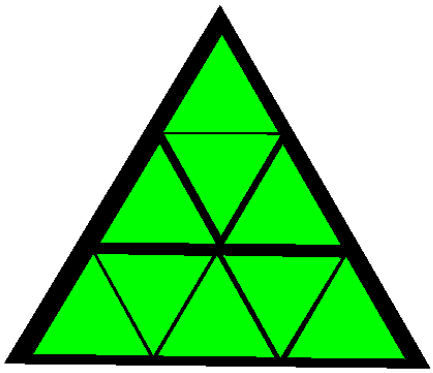
```
870
871         //En medio invertido
872         model = glm::mat4(1.0f);
873         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
874         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
875         model = glm::translate(model, glm::vec3(0.0f, -2.01f, -11.1f));
876         model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 1.0f, 0.0f));
877         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
878         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
879         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
880         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
881         color = glm::vec3(1.0f, 1.0f, 0.0f);
882         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
883         meshList[1]->RenderMesh();
884
885         //En medio dereha
886         model = glm::mat4(1.0f);
887         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
888         model = glm::rotate(model, glm::radians(angulo), glm::vec3(1.0f, 1.0f, 1.0f));
889         model = glm::translate(model, glm::vec3(1.6f, -2.01f, -8.6f));
890         model = glm::scale(model, glm::vec3(3.0f, 3.0f, 3.0f));
891         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
892         glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
893         glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
894         color = glm::vec3(1.0f, 1.0f, 0.0f);
895         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
896         meshList[1]->RenderMesh();
897
```

glm::mat4 Cam
Buscar en línea

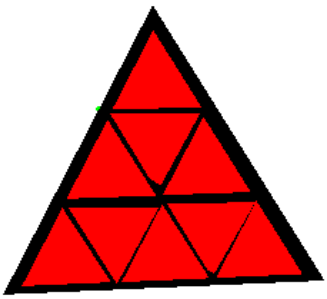
Se agregó la parte de ***angulo+=0.1*** para que todas nuestras figuras puedan rotar a lo largo del programa sin tener que presionar alguna tecla o comando.

```
935         angulo += 0.1;
936         glUseProgram(0);
937         mainWindow.swapBuffers();
938     }
939     return 0;
940 }
941
```

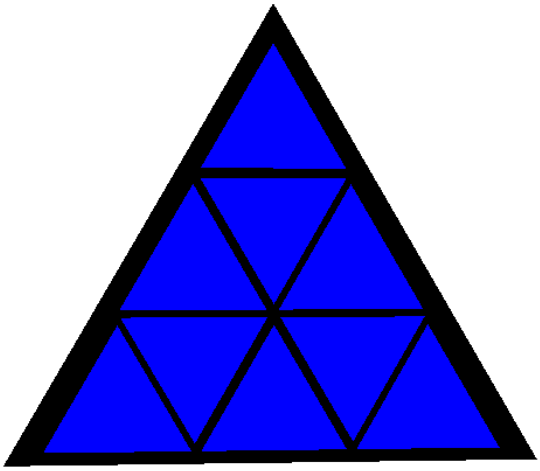
Cara color verde.



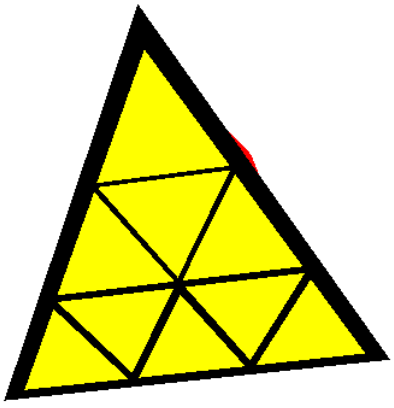
Cara color rojo.



Cara color azul.



Cara color amarillo.



2.- Problemas a la hora de hacer las actividades solicitadas:

Tuve problemas a la hora de andar rotando los triángulos y tratar de que quedaran en la inclinación correcta, dado que la figura no se movía como yo lo imaginaba o quería, y a veces los ejes de rotación se cambiaban cuando se aplicaba algún rotate y eso es considerar ángulos nuevos o que no se pueda hacer todo el giro que quieres de una en un eje, sino que tenemos que hacerla en varias partes

3.- Conclusión:

a. Los ejercicios del reporte: Complejidad, Explicación.
   Creo que el ejercicio fue y es algo desafiante, dado que las pirámides normales son fáciles de acomodar y no es necesario rotarlas, pero con las invertidas se le suma la dificultad de invertirlas mas inclinarlas a una posición en la que queremos para que esta cuadre en la estructura.

b. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias para mejorar desarrollo de la práctica
   Creo que no hubo mucho en lo que faltara explicar o algo, pero si me hubiera gustado tener algún refuerzo o apoyo para poder manejar la inclinación de los triángulos invertidos, si bien esto fue de ayuda para poder aprender mejor a manipularlos, pero me hubiera gustado tener un refuerzo para poder realizar la práctica mejor.

c. Conclusión
   A lo largo de esta práctica pude aprender a realizar instancias de pirámides, así como reforzar lo ya visto con los scale y translate, y el agregado de los rotate. Si bien considero que hay mejora en el resultado de la práctica, pero me ayudó mucho a comprender y manejar las funciones vistas en la práctica.

Bibliografía

- Khronos Group. (2023). OpenGL 4.6 API Core Profile Specification. Recuperado el 1 de marzo de 2025, de https://www.khronos.org/registry/OpenGL/specs/gl/glspec46.core.pdf
- Bailey, M. (2024). Learn OpenGL: Extensive tutorial resource for learning Modern OpenGL. Recuperado el 1 de marzo de 2025, de https://learnopengl.com
- de Vries, J. (2024). Modern OpenGL Tutorials: Transformations and coordinate systems. Recuperado el 1 de marzo de 2025, de https://learnopengl.com/Getting-started/Transformations

- Abi-Chahla, F. (2023). Creating Hierarchical 3D Models with Modern OpenGL and GLM. Recuperado el 1 de marzo de 2025, de https://www.opengl-tutorial.org/intermediate-tutorials/tutorial-17-quaternions/
- Dalmau, D. S. C. (2024). Modelado geométrico y transformaciones jerárquicas con OpenGL y GLM. Revista Digital de Computación Gráfica, 18(2), 45-67. Recuperado el 1 de marzo de 2025, de https://www.revistas.unam.mx/index.php/computacion-grafica/article/view/58942