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 06

NOMBRE COMPLETO: Uriarte Ortiz Enrique Yahir

Nº de Cuenta: 318234757

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 04

SEMESTRE 2025-1

FECHA DE ENTREGA LÍMITE: Sábado 28 de Septiembre del 2024

CALIFICACIÓN: _____

REPORTE DE PRÁCTICA:

1. *Ejecución de los ejercicios.*
   1) Crear un dado de 10 caras y texturizarlo por medio de código.
   Para el desarrollo de esta práctica primero localice las coordenadas en GeoGebra 3D de la base, en un principio había considerado usar un pentágono, pero después de ver imágenes en internet del dado me di cuenta de que tenían una punta inferior en cada cara por lo que decidí que fuera un decágono, y ya que GeoGebra 3D tiene una función para generar polígonos a partir de 2 puntos que defina, la tarea se facilitó más.

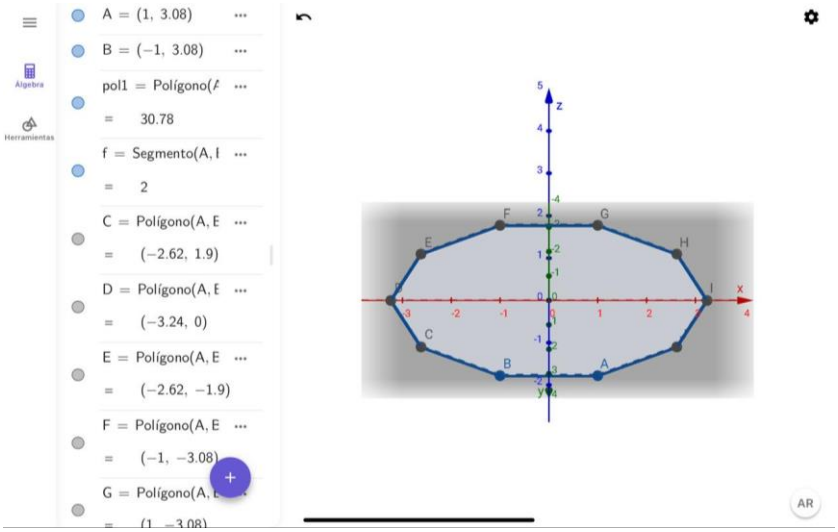

Fig. 1. *Coordenadas de puntos base generadas en GeoGebra 3D.*

Después le di una diferencia de altura de manera salteada a cada punto de la base para dar el efecto que tienen estos dados, por último, definí las coordenadas para las puntas superior e inferior y coloqué rectas de cada punto de la base hacia sus correspondientes puntas. Con lo que ya tenía las coordenadas para el dado y un boceto en 3D de cómo se vería.

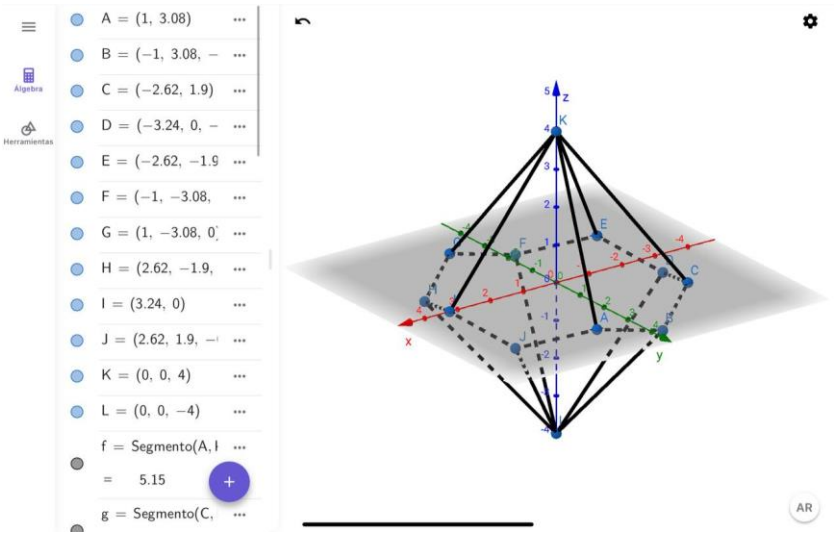

Fig. 2. *Bosquejo en GeoGebra 3D del dado.*

Lo siguiente fue definir las texturas para las caras, utilice las del dado animal, aunque en 3 caras del dado tendría que repetir algunas texturas, decidí que para tener mejor aplicación en cada una, separe cada animal en una imagen aparte, los nombres que les asigne fueron: ***cara_ejefante.tga***, ***cara_pulpo.tga***, ***cara_mariquita.tga***, **cara_vaca.tga** y ***cara_perico-cerdo.tga***, para el caso de las caras con los pericos y cerdos opte por colocar ambos elementos en una sola imagen.

Fig. 3. *Texturizado de cara con imagen de elefante.*

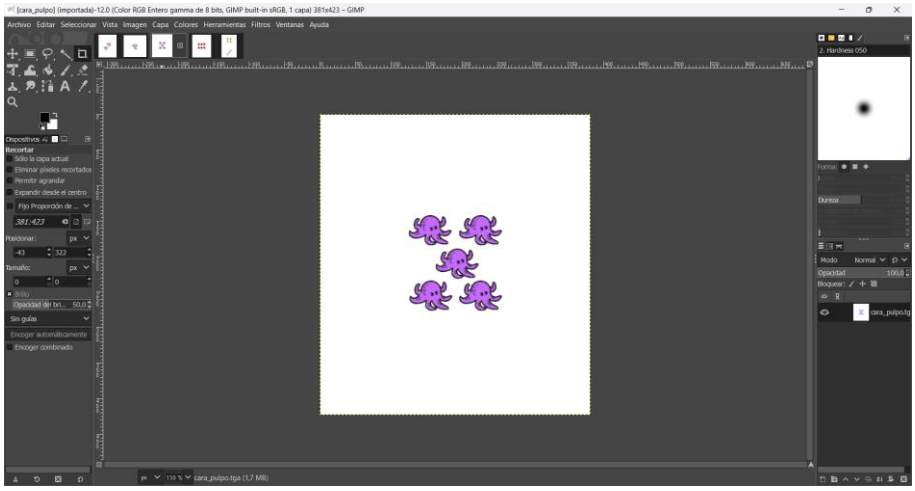

Fig. 4. *Texturizado de cara con imagen de pulpo.*

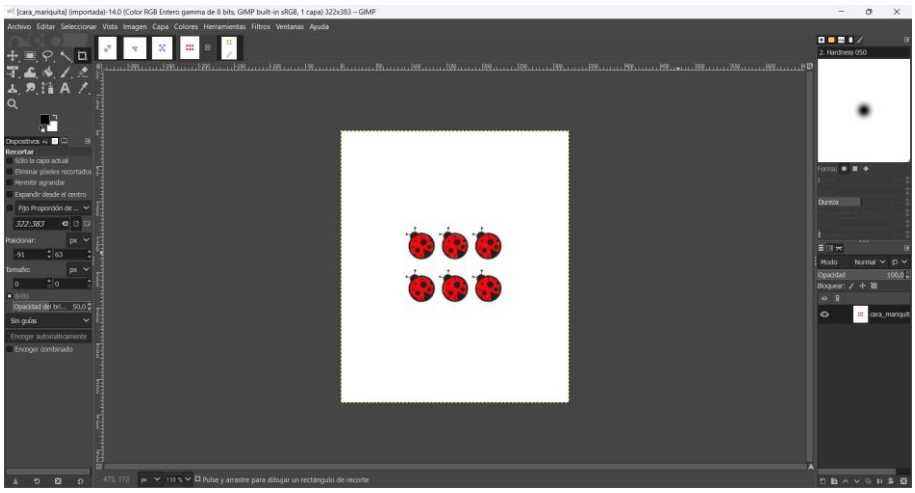

Fig. 5. *Texturizado de cara con imagen de mariquita.*

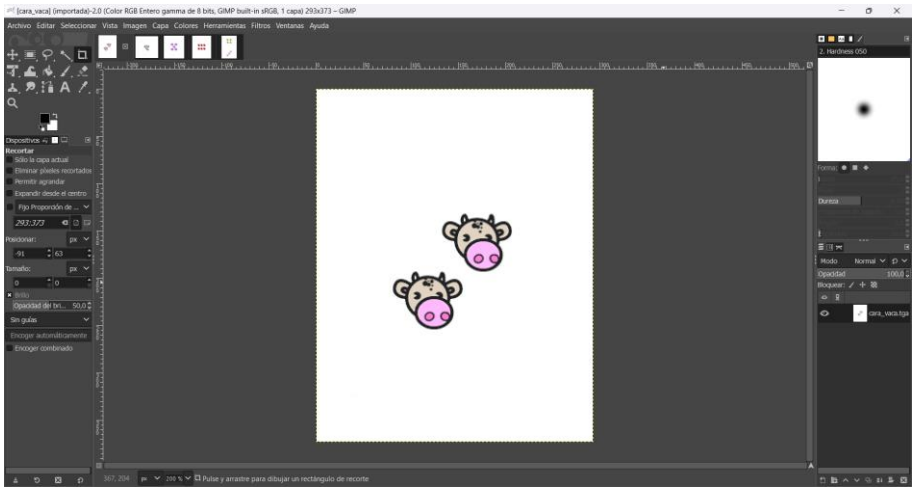

Fig. 6. *Texturizado de cara con imagen de vaca.*

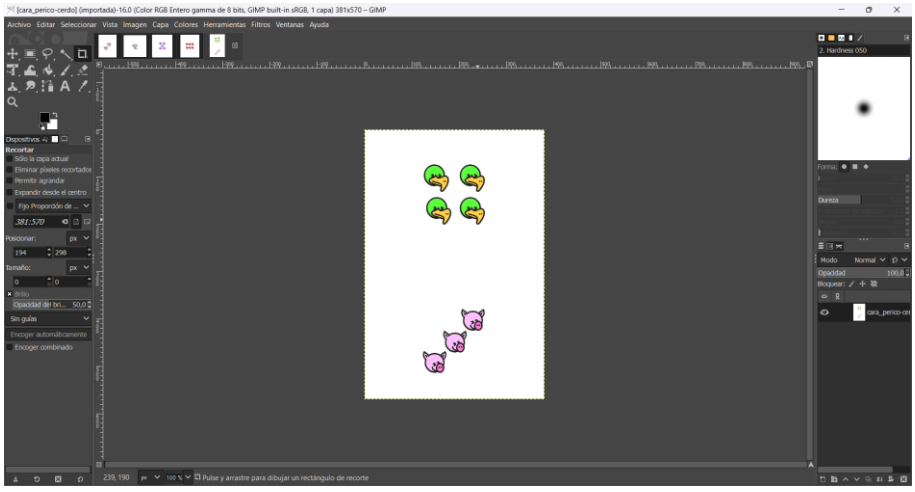

Fig. 7. *Texturizado de cara con imagen de perico y puercos.*

Lo siguiente fue asignar en el código main de la practica las texturas, asignándole un nombre dentro de este y la ubicación dentro de la carpeta para poder utilizarlas.

```
Texture elefanteTexture;
Texture pulpoTexture;
Texture mariquitaTexture;
Texture vacaTexture;
Texture perico_cerdoTexture;
```

Fig. 8. *Definición de Texturas para cada cara del dado.*

```
elefanteTexture = Texture("Textures/cara_ejefante.tga");
elefanteTexture.LoadTextureA();
pulpoTexture = Texture("Textures/cara_pulpo.tga");
pulpoTexture.LoadTextureA();
mariquitaTexture = Texture("Textures/cara_mariquita.tga");
mariquitaTexture.LoadTextureA();
vacaTexture = Texture("Textures/cara_vaca.tga");
vacaTexture.LoadTextureA();
perico_cerdoTexture = Texture("Textures/cara_perico-cerdo.tga");
perico_cerdoTexture.LoadTextureA();
```

Fig. 9. *Definición de localización de archivos para las Texturas.*

Después cree funciones para cada 2 caras, pues este es el numero de caras que utilizaran cada una de las 5 texturas generadas, las caras fueron elegidas al azar y en los puntos de cada cara coloque que coordenada serian de acuerdo con lo que GeoGebra género. Para las texturas las acomodé tras pruebas en ejecución hasta que encontré la ubicación en la que se veían mejor.

```
void CrearDado_CarasElefante(){
    unsigned int cubo_indices[] = {
        0,1,3,      1,2,3,  //Cara 1
        4,5,7,      5,6,7,  //Cara 9
    };
    GLfloat cubo_vertices[] = {
        //Cara 1
        1.00f, 0.00f, 3.08f,    0.25f, 0.30f,   0.00f, 0.00f,-1.00f,//A - 0
       -1.00f,-0.50f, 3.08f,    0.50f, 0.20f,   0.00f, 0.00f,-1.00f,//B - 1
       -2.62f, 0.00f, 1.90f,    0.75f, 0.30f,   0.00f, 0.00f,-1.00f,//C - 2
        0.00f, 4.00f, 0.00f,    0.50f, 1.00f,   0.00f, 0.00f,-1.00f,//K - 3
        //Cara 9
        2.62f,-0.50f,-1.90f,    0.25f, 0.70f,   0.00f, 0.00f,-1.00f,//H - 4
        3.24f, 0.00f, 0.00f,    0.50f, 0.80f,   0.00f, 0.00f,-1.00f,//I - 5
        2.62f,-0.50f, 1.90f,    0.75f, 0.70f,   0.00f, 0.00f,-1.00f,//J - 6
        0.00f,-4.50f, 0.00f,    0.50f, 0.00f,   0.00f, 0.00f,-1.00f,//L - 7
    };
    Mesh* dado = new Mesh();
    dado->CreateMesh(cubo_vertices, cubo_indices, 64, 12);
    meshList.push_back(dado);}
```

Fig. 10. *Coordenadas de Caras 1 y 9, con texturas ajustadas.*

```
void CrearDado_CarasPulpo() {
    unsigned int cubo_indices[] = {
        0,1,3,      2,1,3,  //Cara 2
        4,5,7,      5,6,7,  //Cara 7
    };
    GLfloat cubo_vertices[] = {
        //Cara 2
       -2.62f, 0.00f, 1.90f,    0.13f, 0.35f,   0.00f, 0.00f,-1.00f,//C - 0
       -3.24f,-0.50f, 0.00f,    0.50f, 0.25f,   0.00f, 0.00f,-1.00f,//D - 1
       -2.62f, 0.00f,-1.90f,    0.87f, 0.35f,   0.00f, 0.00f,-1.00f,//E - 2
        0.00f, 4.00f, 0.00f,    0.50f, 1.00f,   0.00f, 0.00f,-1.00f,//K - 3
        //Cara 7
       -3.24f,-0.50f, 0.00f,    0.13f, 0.65f,   0.00f, 0.00f,-1.00f,//D - 4
       -2.62f, 0.00f,-1.90f,    0.50f, 0.75f,   0.00f, 0.00f,-1.00f,//E - 5
       -1.00f,-0.50f,-3.08f,    0.87f, 0.65f,   0.00f, 0.00f,-1.00f,//F - 6
        0.00f,-4.50f, 0.00f,    0.50f, 0.00f,   0.00f, 0.00f,-1.00f,//L - 7
    };
    Mesh* dado = new Mesh();
    dado->CreateMesh(cubo_vertices, cubo_indices, 64, 12);
    meshList.push_back(dado);
}
```

Fig. 11. *Coordenadas de Caras 2 y 7, con texturas ajustadas.*

```
void CrearDado_CarasMariquita() {
    unsigned int cubo_indices[] = {
        0,1,3,      2,1,3,  //Cara 3
        4,5,7,      5,6,7,  //Cara 10
    };
    GLfloat cubo_vertices[] = {
        //Cara 3
       -2.62f, 0.00f,-1.90f,    0.10f, 0.35f,   0.00f, 0.00f,-1.00f,//E - 0
       -1.00f,-0.50f,-3.08f,    0.50f, 0.25f,   0.00f, 0.00f,-1.00f,//F - 1
        1.00f, 0.00f,-3.08f,    0.90f, 0.35f,   0.00f, 0.00f,-1.00f,//G - 2
        0.00f, 4.00f, 0.00f,    0.50f, 1.00f,   0.00f, 0.00f,-1.00f,//K - 3
        //Cara 10
        2.62f,-0.50f, 1.90f,    0.10f, 0.65f,   0.00f, 0.00f,-1.00f,//J - 4
        1.00f, 0.00f, 3.08f,    0.50f, 0.75f,   0.00f, 0.00f,-1.00f,//A - 5
       -1.00f,-0.50f, 3.08f,    0.90f, 0.65f,   0.00f, 0.00f,-1.00f,//B - 6
        0.00f,-4.50f, 0.00f,    0.50f, 0.00f,   0.00f, 0.00f,-1.00f,//L - 7
    };
    Mesh* dado = new Mesh();
    dado->CreateMesh(cubo_vertices, cubo_indices, 64, 12);
    meshList.push_back(dado);
}
```

Fig. 12. *Coordenadas de Caras 3 y 10, con texturas ajustadas.*

```
void CrearDado_CarasVaca() {
    unsigned int cubo_indices[] = {
        0,1,3,      2,1,3,  //Cara 4
        4,5,7,      5,6,7,  //Cara 8
    };
    GLfloat cubo_vertices[] = {
        //Cara 4
        1.00f, 0.00f,-3.08f,    0.10f, 0.35f,   0.00f, 0.00f,-1.00f,//G - 0
        2.62f,-0.50f,-1.90f,    0.50f, 0.25f,   0.00f, 0.00f,-1.00f,//H - 1
        3.24f, 0.00f, 0.00f,    0.90f, 0.35f,   0.00f, 0.00f,-1.00f,//I - 2
        0.00f, 4.00f, 0.00f,    0.50f, 1.00f,   0.00f, 0.00f,-1.00f,//K - 3
        //Cara 8
       -1.00f,-0.50f,-3.08f,    0.10f, 0.65f,   0.00f, 0.00f,-1.00f,//F - 4
        1.00f, 0.00f,-3.08f,    0.50f, 0.75f,   0.00f, 0.00f,-1.00f,//G - 5
        2.62f,-0.50f,-1.90f,    0.90f, 0.65f,   0.00f, 0.00f,-1.00f,//H - 6
        0.00f,-4.50f, 0.00f,    0.50f, 0.00f,   0.00f, 0.00f,-1.00f,//L - 7
    };
    Mesh* dado = new Mesh();
    dado->CreateMesh(cubo_vertices, cubo_indices, 64, 12);
    meshList.push_back(dado);
}
```

Fig. 13. *Coordenadas de Caras 4 y 8, con texturas ajustadas.*

```
void CrearDado_Caras_PC() {
    unsigned int cubo_indices[] = {
        0,1,3,      2,1,3,  //Cara 5
        4,5,7,      5,6,7,  //Cara 6
    };
    GLfloat cubo_vertices[] = {
        //Cara 5
        3.24f, 0.00f, 0.00f,    0.10f, 0.625f,  0.00f, 0.00f,-1.00f,//I - 0
        2.62f,-0.50f, 1.90f,    0.50f, 0.58f,   0.00f, 0.00f,-1.00f,//J - 1
        1.00f, 0.00f, 3.08f,    0.90f, 0.625f,  0.00f, 0.00f,-1.00f,//A - 2
        0.00f, 4.00f, 0.00f,    0.50f, 1.10f,   0.00f, 0.00f,-1.00f,//K - 3
        //Cara 6
       -1.00f,-0.50f, 3.08f,    0.05f, 0.375f,  0.00f, 0.00f,-1.00f,//B - 4
       -2.62f, 0.00f, 1.90f,    0.45f, 0.46f,   0.00f, 0.00f,-1.00f,//C - 5
       -3.24f,-0.50f, 0.00f,    0.85f, 0.375f,  0.00f, 0.00f,-1.00f,//D - 6
        0.00f,-4.50f, 0.00f,    0.45f,-0.10f,   0.00f, 0.00f,-1.00f,//L - 7
    };
    Mesh* dado = new Mesh();
    dado->CreateMesh(cubo_vertices, cubo_indices, 64, 12);
    meshList.push_back(dado);
}
```

Fig. 14. *Coordenadas de Caras 5 y 6, con texturas ajustadas.*

En la función main defino las funciones para cada una de las caras del dado.

```
CrearDado_CarasElefante();
CrearDado_CarasPulpo();
CrearDado_CarasMariquita();
CrearDado_CarasVaca();
CrearDado_Caras_PC();
```

Fig. 15. *Declaración de funciones para cada cara.*

Y por último mando a llamar cada función con su respectiva textura ajustada en posición y escala.

```
//Ejercicio 1: Crear un dado de 10 caras y texturizarlo por medio de código.

model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, 2.0f, -4.0f));
model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
elefanteTexture.UseTexture();
meshList[4]->RenderMesh();
pulpoTexture.UseTexture();
meshList[5]->RenderMesh();
mariquitaTexture.UseTexture();
meshList[6]->RenderMesh();
vacaTexture.UseTexture();
meshList[7]->RenderMesh();
perico_cerdoTexture.UseTexture();
meshList[8]->RenderMesh();
```

Fig. 16. *Llamada de impresión del dado.*

2) Importar el modelo de su coche con sus 4 llantas acomodadas y tener texturizadas las 4 llantas (diferenciar caucho y rin).
Para esta actividad obtuve 2 imágenes de internet de una llanta, una donde se veía el cucho y el rin, y otra donde se ve el grabado del caucho, en GIMP las junté en una sola imagen, importándola como ***Llantas_Carro.tga***.

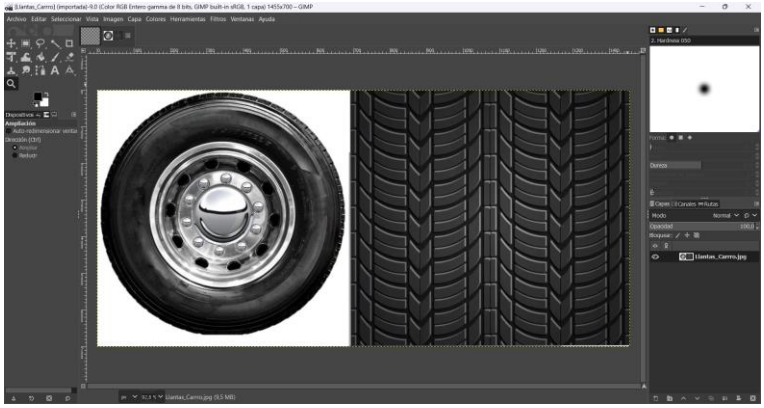

Fig. 17. *Ajuste de imagen en GIMP.*

En 3ds Max asigne las texturas y las acomode a cada cara del modelo de la llanta, la cara donde estaría el rin y el caucho, y la parte donde estaría solo el caucho con el grabado.

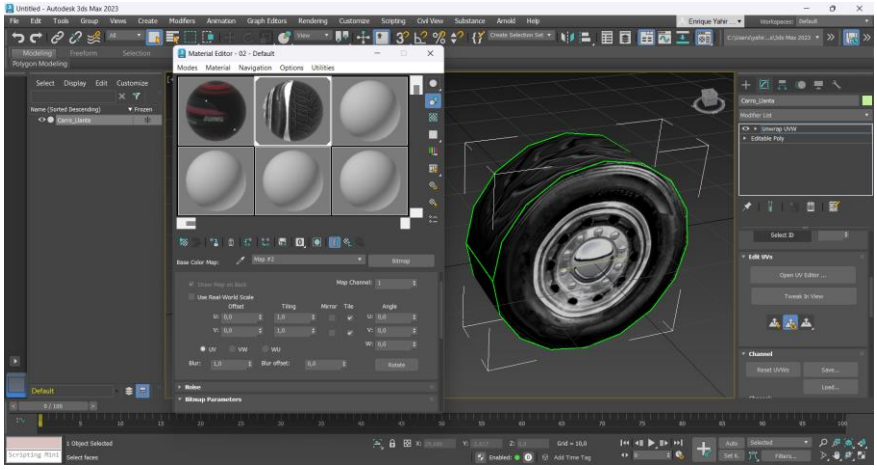

Fig. 18. *Ajuste de textura en 3ds Max.*

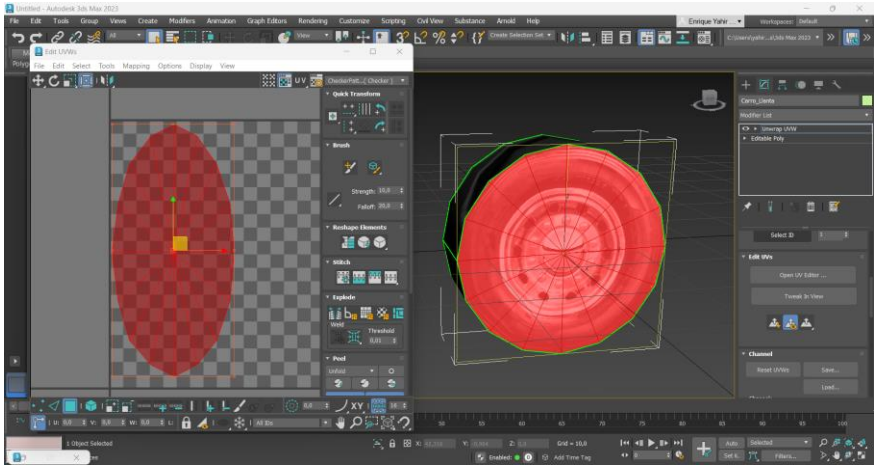

Fig. 19. *Ajuste de textura en 3ds Max.*

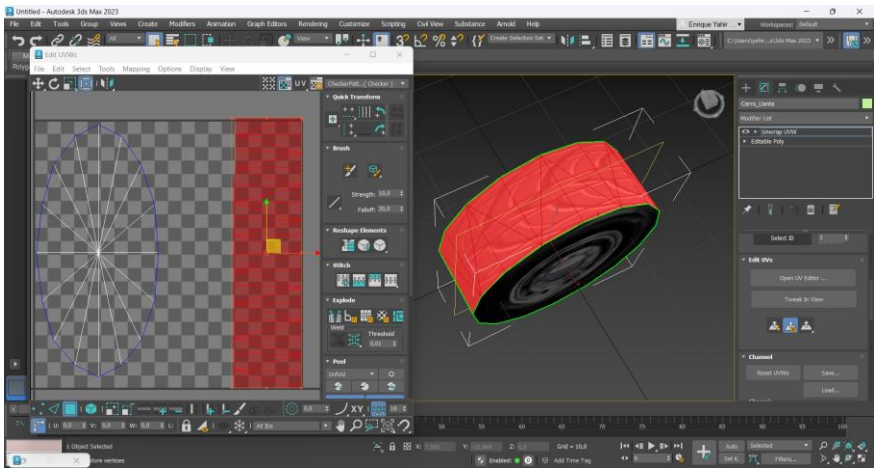

Fig. 20. *Ajuste de textura en 3ds Max.*

Después asigne en el código main de la practica el modelo de la llanta con texturas, además de los modelos del cofre y el cuerpo del carro, pero estos sin texturas definidas, dándole un nombre a cada uno y la ubicación dentro de la carpeta para poder utilizarlos.

```
Model Carro_Llantas;
Model Carro_Cuerpo;
Model Carro_Cofre;
```

Fig. 21. *Definición de Modelos para cada parte del carro.*

```
Carro_Cuerpo = Model();
Carro_Cuerpo.LoadModel("Models/Carro_Cuerpo.obj");
Carro_Cofre = Model();
Carro_Cofre.LoadModel("Models/Carro_Cofre.obj");
Carro_Llantas = Model();
Carro_Llantas.LoadModel("Models/Carro_Llanta.obj");
```

Fig. 22. *Definición de localización de archivos para los modelos.*

```
model = glm::mat4(1.0);//Cuerpo Carro
model = glm::translate(model, glm::vec3(0.0f, 0.0f, 3.0f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::scale(model, glm::vec3(0.05f, 0.05f, 0.05f));
Cofre = model;
Llanta01 = model;
Llanta02 = model;
Llanta03 = model;
Llanta04 = model;
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Carro_Cuerpo.RenderModel();

model = Llanta01;//Llanta01. Derecha.
model = glm::translate(model, glm::vec3(30.0f, -20.0f, 70.0f));
Llanta01 = model;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Carro_Llantas.RenderModel();

model = Llanta02;//Llanta02. Izquierda.
model = glm::translate(model, glm::vec3(-30.0f, -20.0f, 70.0f));
model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 1.0f, 0.0f));
Llanta02 = model;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Carro_Llantas.RenderModel();
```

Fig. 23. *Llamada de impresión del carro.*

Por último, mando a llamar cada parte del carro para mostrarlo en la ejecución, ajustando su tamaño y posición, la estructura del código es la misma que el de la practica anterior en esta parte, solo que elimine las líneas de código en cada modelo para poder rotarlos con una tecla, pero las jerarquías son las mismas.
Así mismo en esta ocasión solo utilice un modelo para todas las llantas, y para las llantas que debían tener la vista del rin del lado contrario solo les agregue la instrucción de rotación a 180° en Y.

```
model = Llanta03;//Llanta03. Izquierda.
model = glm::translate(model, glm::vec3(-30.0f, -21.0f, -55.0f));
model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 1.0f, 0.0f));
Llanta03 = model;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Carro_Llantas.RenderModel();

model = Llanta04;//Llanta04. Derecha.
model = glm::translate(model, glm::vec3(30.0f, -21.0f, -55.0f));
Llanta04 = model;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Carro_Llantas.RenderModel();

model = Cofre;//Cofre de carro.
model = glm::translate(model, glm::vec3(0.0f, 11.0f, 50.0f));
Cofre = model;
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Carro_Cofre.RenderModel();
```

Fig. 24. *Llamada de impresión del carro.*

3) Texturizar la cara del personaje de la imagen tipo cars en el espejo (ojos) y detalles en cofre y parrilla de su propio modelo de coche.
En esta actividad obtuve as texturas para las partes que se pedían después de buscar en internet, pero la parte del cofre era complicada de encontrar ya que en la imagen no se apreciaba muy bien, por lo que opté por buscar las texturas del personaje de otro modelo 3D y conseguí la que tenía el cofre. En GIMP ajuste para recorte únicamente la parte del cofre y la guarde como ***cofre.tga***.

Fig. 25. *Imagen de Texturas conseguida para el cofre.*

Para el frente y la parrilla si encontré una imagen de frente del personaje del que podía extraer la textura, al igual que el cofre en GIMP la ajuste, recortando los ojos y la parte del frente, pero como había unos detalles más en lo que quedo del cofre del carro, por lo que también agregue en esta la textura del cofre, la guarde como ***Frente.tga***.
En 3ds Max ajuste las dimensiones y coloque las texturas en cada modelo.

Fig. 26. *Imagen de Texturas conseguida para espejo y parrilla.*

Fig. 27. *Ajuste de imagen en GIMP del cofre.*

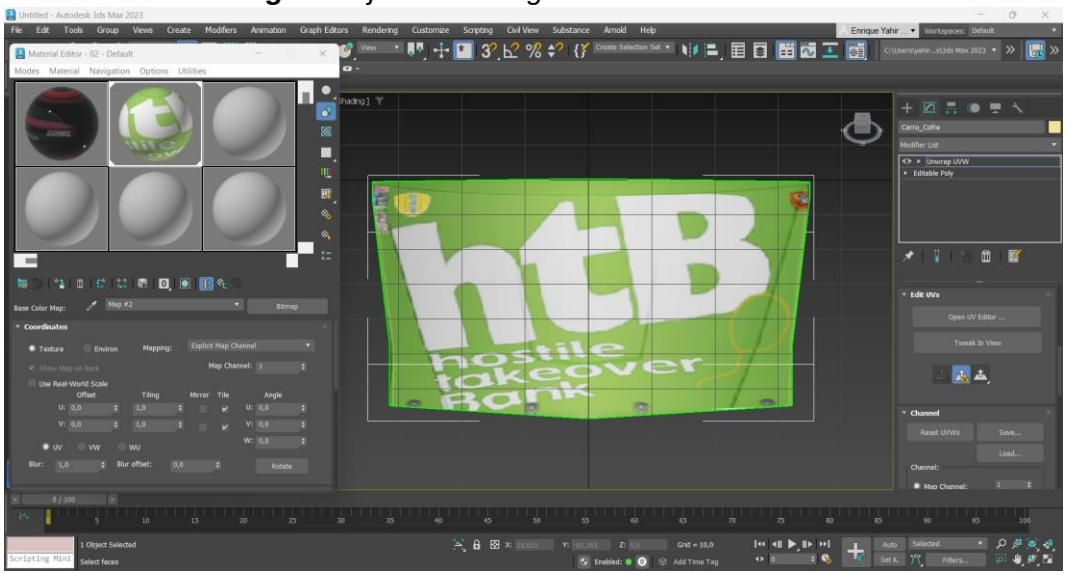

Fig. 28. *Ajuste de textura en 3ds Max.*

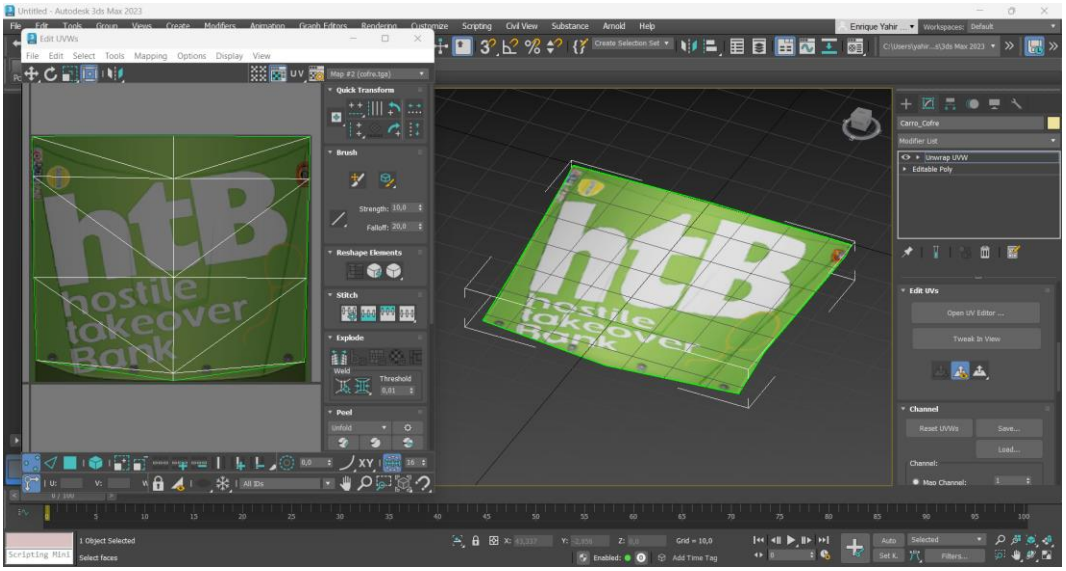

Fig. 29. *Ajuste de textura en 3ds Max.*

Fig. 30. *Ajuste de imagen en GIMP del frente.*

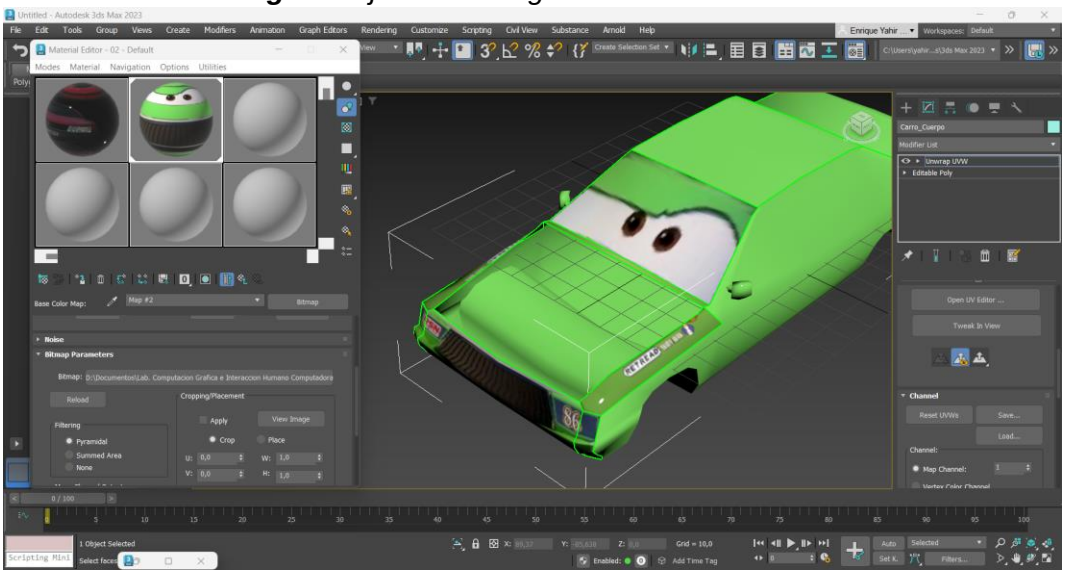

Fig. 31. *Ajuste de textura en 3ds Max.*

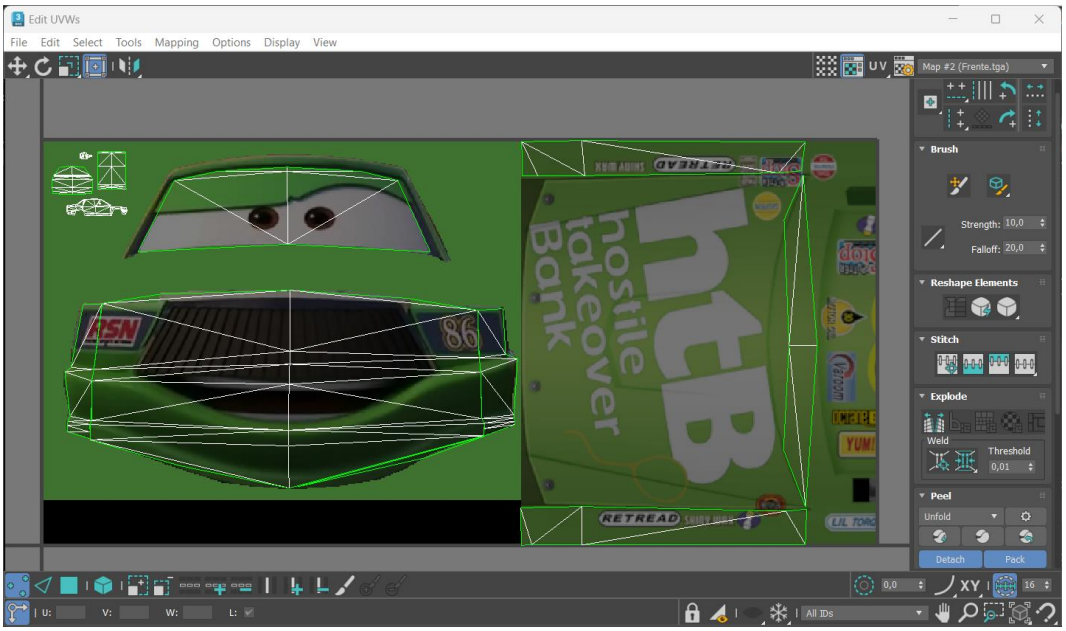

Fig. 32. *Ajuste de textura en 3ds Max.*

Fig. 33. *Ajuste de textura en 3ds Max.*

En esta actividad no se modificó el código main, únicamente se necesitaba colocar las texturas en el carro.

La ejecución de las 3 actividades juntas fue la siguiente:

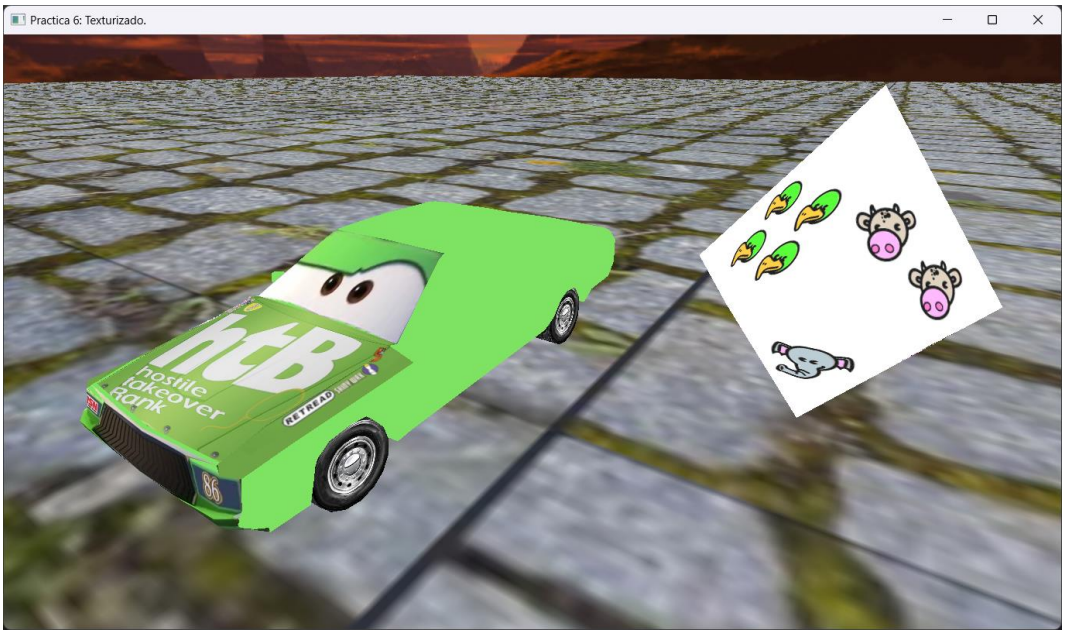

Fig. 24. *Ejecución del código.*

Como evidencia para la práctica, realice un video donde se muestra la ejecución, el archivo se llama P06-318234757_Ejercicios.mp4, adjunto el enlace de la carpeta compartida:https://drive.google.com/file/d/1QYqPTcPKpJpUwwSRPobrf701wUBvt0bE/view?usp=drive_link

2. *Problemas presentados.*
   En las primeras 3 actividades no presente problemas muy grabes, las 3 tuvieron su nivel de complicación, aunque en la segunda actividad cuando exportaba el modelo en .obj de la llanta al Visual Studio, al momento de ejecutar no me respetaba las texturas y se mostraba de color negro, después de unos intentos buscando la causa de este error había considerado que quizás era el tipo de imagen, pero en GIMP la había configurado como la textura del dado. Decidí probar con las texturas del dado colocadas en la llanta y esas si las exportaba en la ejecución, incluso también probe con la imagen del escudo de la fi en la llanta y también la respetaba, al final me di cuenta que debía poner la textura también en la carpeta ***Textures***, y solo así al momento de ejecutar el programa si me aparecieron las texturas para la llanta, esto me sirvió para estar prevenido en la siguiente actividad donde debía también exportar las texturas en las demás partes del carro. De ahí en fuera no presente más problemas.

3. *Conclusión:*
   Tras realizar los ejercicios de la practica me parece que la complejidad era normal, la implementación de texturizado tanto por código como por programa de edición me parecieron muy interesantes, aunque me gusto mas el texturizado por 3ds Max ya que es más preciso de controlar, en cuanto a la estructura de modelos del carro fue copiar lo realizado en la practica anterior, lo cual no se me hizo para nada complicado. En cuanto al texturizado por código se me hizo más laborioso por los múltiples intento que debía hacer para ajustar las caras. Al final consideró que esta práctica cumplió su

objetivo de introducirnos al uso de texturas en nuestros modelos, para poder combinar o cambiar los texturizados si es que no nos gusta el que viene en el modelo, siendo esto a mi parecer una buena práctica para el desarrollo del proyecto final.

4. *Bibliografía.*

- Wii - Cars - Chick Hicks. The Models Resource. https://www.models-resource.com/wii/cars/model/65519/ .Accedido septiembre 26, 2024.
- Nyberg P. Llanta camión cromada. Pinterest. https://pin.it/6FjOzakFB. Publicado el 10 de enero de 2022. Accedido septiembre 26, 2024.
- Wheel tire seamless pattern. Depositphotos. https://depositphotos.com/vector/wheel-tire-seamless-pattern-52484679.html Accedido septiembre 26, 2024.
- PaddyMcClellan. Chick Hicks Png. DeviantArt. https://www.deviantart.com/paddymcclellan/art/Chick-Hicks-Png-1011681939. Publicado el 16 de enero de 2024. Accedido septiembre 26, 2024.