



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° 6

NOMBRE COMPLETO: Uriarte Ortiz Enrique Yahir

N° de Cuenta: 318234757

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 04

SEMESTRE 2025-1

FECHA DE ENTREGA LÍMITE: Martes 24 de Septiembre del 2024

CALIFICACIÓN: _____

EJERCICIOS DE SESIÓN:

1. Actividades realizadas.

1) Texturizar su cubo con la imagen dado_animales ya optimizada por ustedes.

Para esta actividad lo primero ajustar la imagen en GIMP de acuerdo como lo mostro el profesor, recortando la imagen y ajustando las dimensiones de pixeles a 512 x 512, para después guardarlo como dado_animales.tga.

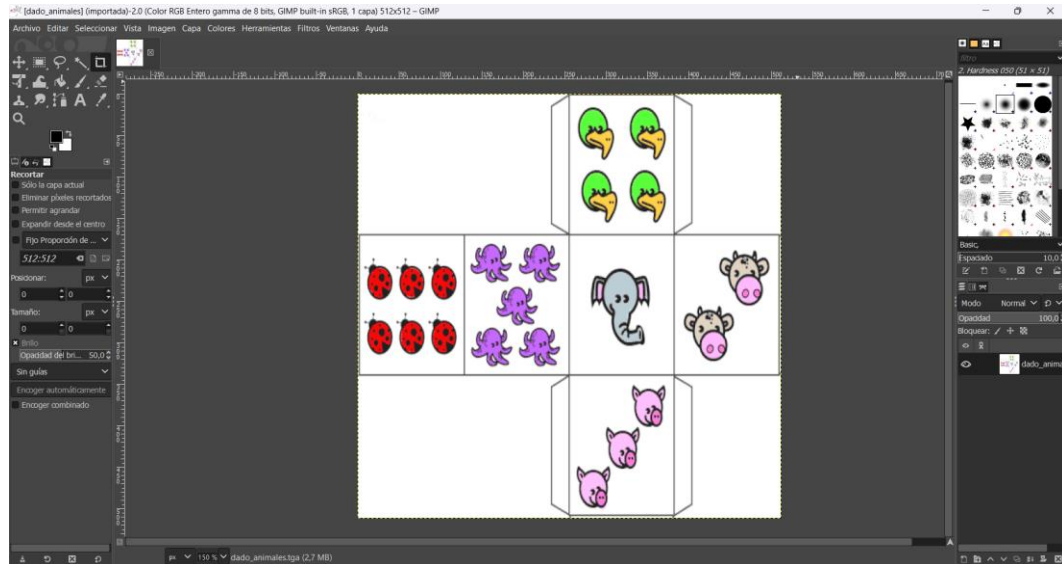


Fig. 1. Ajuste de la imagen del dado de animales.

Ya en el archivo main del proyecto declaro el nombre de las texturas.

```
Texture dadoTexture;
```

Fig. 2. Declaración de texturas.

Después en la función para declarar las posiciones de las texturas en cada cara del cubo, ajuste en las coordenadas correspondientes a S y T a las áreas donde se encontraban las caras del cubo, por lo general el ancho de cada cubo fue de 0.23 a 0.24, mientras que de altura fue de 0.3 a 0.32.

```
void CrearDado(){
    unsigned int cubo_indices[] = { ... }

    GLfloat cubo_vertices[] = { // average normals
        //x    y    z    S    T    NX    NY    NZ
        // front
        -0.5f, -0.5f, 0.5f, 0.26f, 0.34f, 0.0f, 0.0f, -1.0f, //0
        0.5f, -0.5f, 0.5f, 0.49f, 0.34f, 0.0f, 0.0f, -1.0f, //1
        0.5f, 0.5f, 0.5f, 0.49f, 0.66f, 0.0f, 0.0f, -1.0f, //2
        -0.5f, 0.5f, 0.5f, 0.26f, 0.66f, 0.0f, 0.0f, -1.0f, //3
        // right
        0.5f, -0.5f, 0.5f, 0.5f, 0.34f, -1.0f, 0.0f, 0.0f,
        0.5f, -0.5f, -0.5f, 0.74f, 0.34f, -1.0f, 0.0f, 0.0f,
        0.5f, 0.5f, -0.5f, 0.74f, 0.66f, -1.0f, 0.0f, 0.0f,
        0.5f, 0.5f, 0.5f, 0.5f, 0.66f, -1.0f, 0.0f, 0.0f,
```

Fig. 3. Ajuste de caras del cubo.

```

// back
-0.5f, -0.5f, -0.5f, 0.99f, 0.34f, 0.0f, 0.0f, 1.0f,
0.5f, -0.5f, -0.5f, 0.75f, 0.34f, 0.0f, 0.0f, 1.0f,
0.5f, 0.5f, -0.5f, 0.75f, 0.66f, 0.0f, 0.0f, 1.0f,
-0.5f, 0.5f, -0.5f, 0.99f, 0.66f, 0.0f, 0.0f, 1.0f,
// left
-0.5f, -0.5f, -0.5f, 0.01f, 0.34f, 1.0f, 0.0f, 0.0f,
-0.5f, -0.5f, 0.5f, 0.25f, 0.34f, 1.0f, 0.0f, 0.0f,
-0.5f, 0.5f, 0.5f, 0.25f, 0.66f, 1.0f, 0.0f, 0.0f,
-0.5f, 0.5f, -0.5f, 0.01f, 0.66f, 1.0f, 0.0f, 0.0f,
// bottom
0.5f, -0.5f, 0.5f, 0.5f, 0.32f, 0.0f, 1.0f, 0.0f,
0.5f, -0.5f, -0.5f, 0.74f, 0.32f, 0.0f, 1.0f, 0.0f,
-0.5f, -0.5f, -0.5f, 0.74f, 0.01f, 0.0f, 1.0f, 0.0f,
-0.5f, -0.5f, 0.5f, 0.5f, 0.01f, 0.0f, 1.0f, 0.0f,
//UP
0.5f, 0.5f, 0.5f, 0.5f, 0.68f, 0.0f, -1.0f, 0.0f,
0.5f, 0.5f, -0.5f, 0.74f, 0.68f, 0.0f, -1.0f, 0.0f,
-0.5f, 0.5f, -0.5f, 0.74f, 0.98f, 0.0f, -1.0f, 0.0f,
-0.5f, 0.5f, 0.5f, 0.5f, 0.98f, 0.0f, -1.0f, 0.0f,};

Mesh* dado = new Mesh();
dado->CreateMesh(cubo_vertices, cubo_indices, 192, 36);
meshList.push_back(dado);}

```

Fig. 4. Ajuste de caras del cubo.

Luego declare el archivo que se usara para las texturas del cubo.

```

dadoTexture = Texture("Texturas/dado_animales.tga");
dadoTexture.LoadTextureA();

```

Fig. 5. Texturas para el cubo.

Por último, declare la impresión del cubo, ajuste la translación para que no se encontrara tan alto y le agregue una escala para que fuera un poco mas grande pero no tanto como el cubo importado de 3ds Max en la siguiente actividad.

```

model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, 1.5f, -2.0f));
model = glm::scale(model, glm::vec3(1.25f, 1.25f, 1.25f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
dadoTexture.UseTexture();
meshList[4]->RenderMesh();

```

Fig. 6. Impresión de cubo imagen optimizada.

- 2) Importar el cubo texturizado en el programa de modelado con la imagen dado_animales ya optimizada por ustedes.

Para esta segunda actividad cree desde cero el cubo, añadir las texturas a cada cara con ayuda de la opción de “bittmap” y con “Open UV Editor” ajustar la sección de la textura para cada cara.

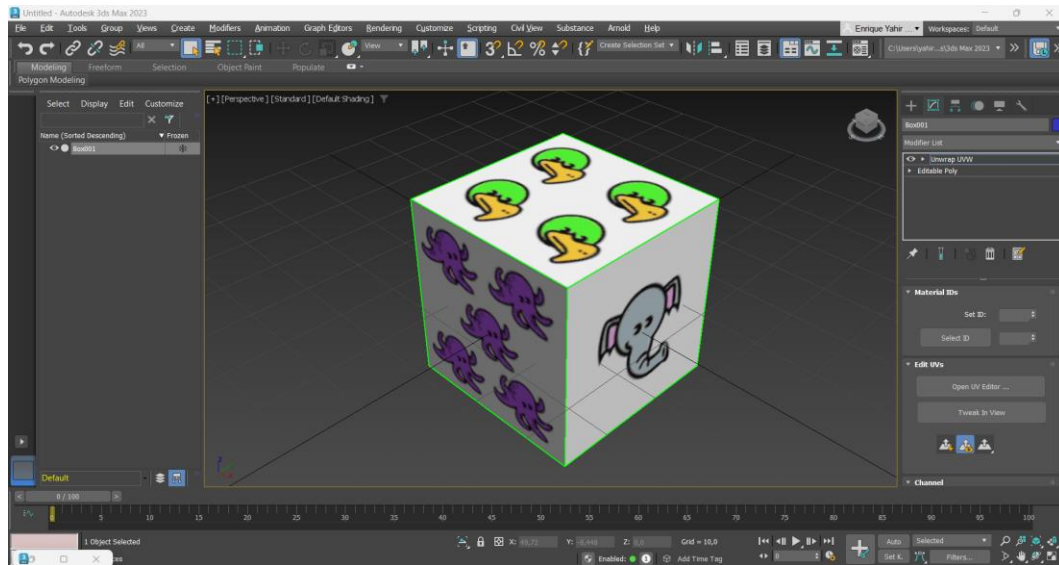


Fig. 7. Creación de dado en 3ds Max con imagen de dado_animales.

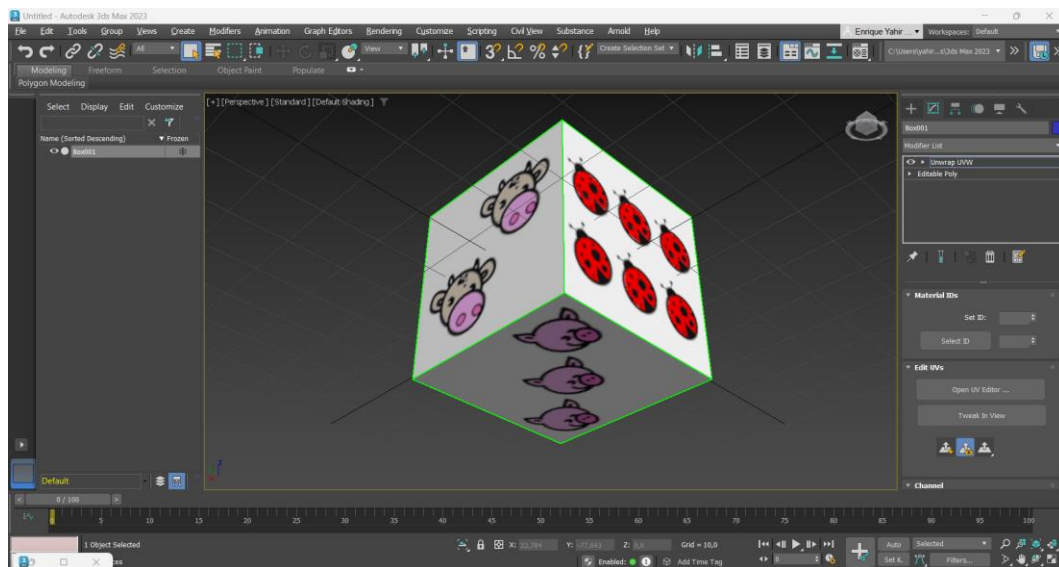


Fig. 8. Creación de dado en 3ds Max con imagen de dado_animales.

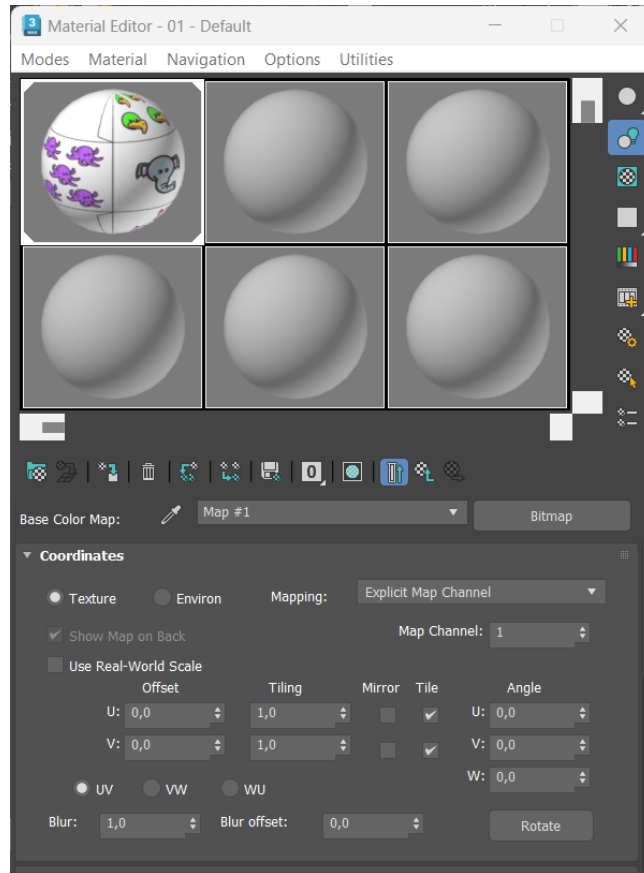


Fig. 9. Textura de Material para las caras del dado animales.

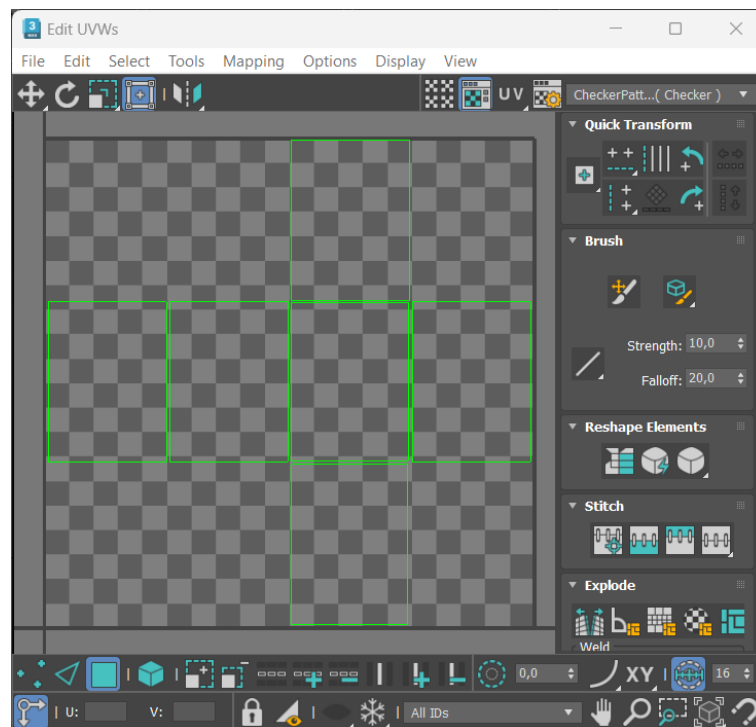


Fig. 10. Ajuste de sección para cada cara del cubo.

Después lo exporte como archivo tipo “.obj” cargado con sus texturas, lo que hace el archivo “.mtl” con el la imagen de los dados que es “.tga”.

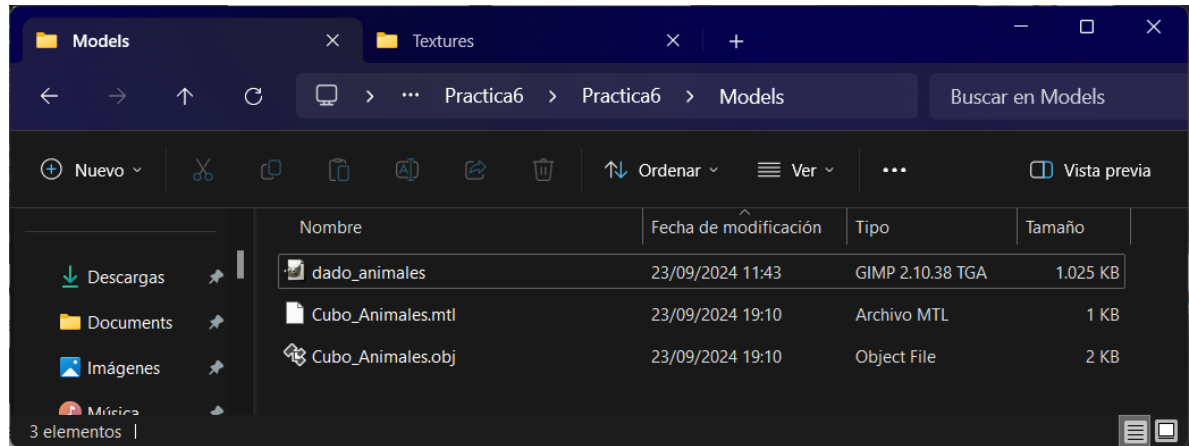


Fig. 11. Archivos del cubo generados.

Ya en el archivo main del proyecto declaro el nombre del modelo del dado.

```
Model Dado_M;
```

Fig. 12. Declaración de modelo.

Luego declare la ubicación del modelo.

```
Dado_M = Model();  
Dado_M.LoadModel("Models/Cubo_Animales.obj");
```

Fig. 13. Importación de modelo con texturas.

Por último, declare la impresión del modelo y ajuste la translación para que no se encontrara tan alto.

```
model = glm::mat4(1.0f);  
model = glm::translate(model, glm::vec3(-3.0f, 1.5f, -2.0f));  
model = glm::scale(model, glm::vec3(0.05f, 0.05f, 0.05f));  
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));  
Dado_M.RenderModel();
```

Fig. 14. Impresión de modelo de dado.

En la ejecución final de código el dado del lado derecho es el de la actividad 1, mientras que el de la izquierda es el de la actividad 2, una forma de diferenciarlos es que el dado de la actividad 1 es mas pequeño que el de la actividad 2.

Ejecucion del programa:



Fig. 15. Ejecución del código.

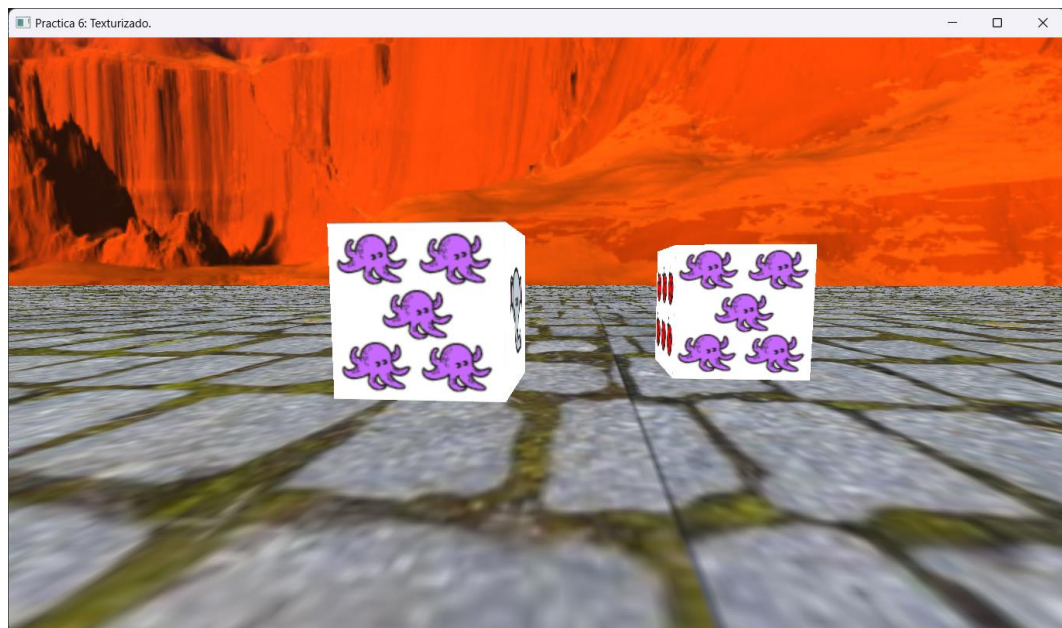


Fig. 16. Ejecución del código.

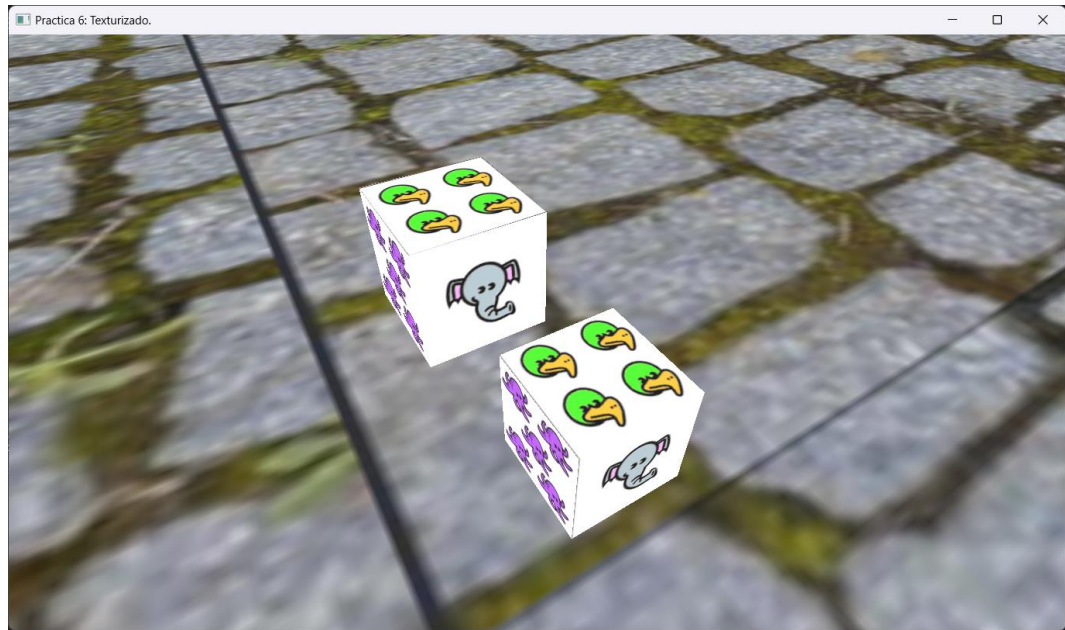


Fig. 17. Ejecución del código.

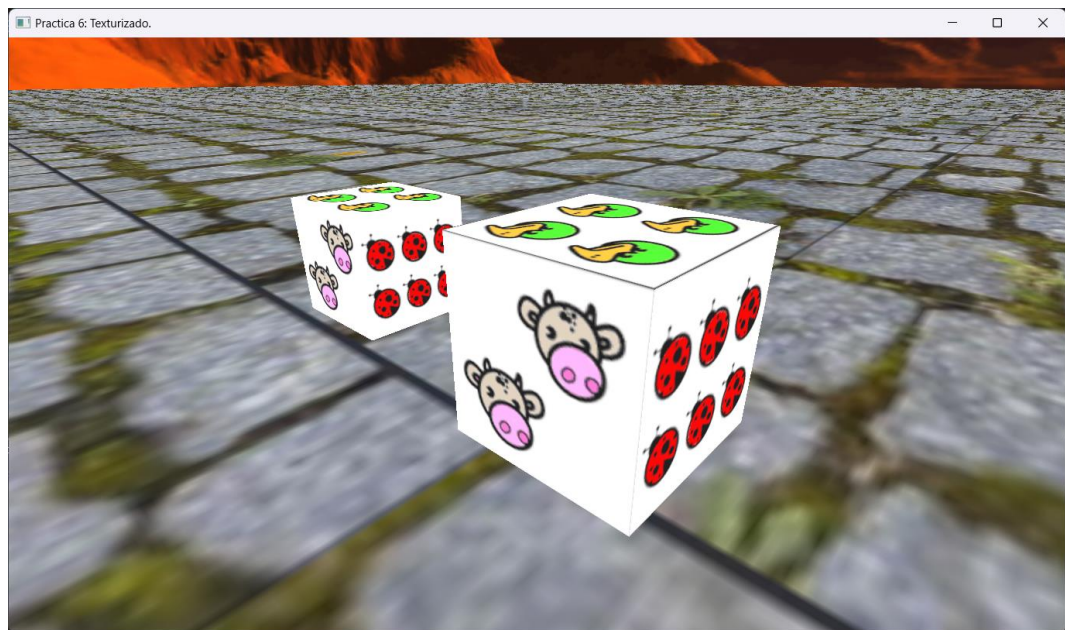


Fig. 18. Ejecución del código.



Fig. 19. Ejecución del código.

2. Problemas presentados.

Para estos ejercicios si presente algunas complicaciones, en el primer ejercicio en la sección del código main donde se declara cada cara, en lo que es la cara back y left modifique el orden de los puntos ya que al ejecutarlos por default las imágenes se encontraban en espejo, mientras que en las caras bottom y up también modifique las coordenadas en X y Y para que las imagen se encontrara con respecto a la cara right que tenía el elefante, por lo que lo complicado del ejercicio lo pude resolver fácilmente. Otra complicación que tuve fue en el ejercicio 2, en la modificación de 3dsMax, por alguna razón cuando realizaba los ajustes de las secciones de cada cara según la imagen no me la mostraba como referencia, pero a pesar de este error logré observar la ventana principal como quedaba cada cara con el ajuste por lo que no me fue tan complicado al final.

De ahí en fuera no presente algún otro error o problema.

3. Conclusión:

Después de realizar el ejercicio propuesto, considero que la complejidad que tenían los ejercicios era normal, la implementación de texturas en los modelos ya sea dentro del código o ya exportando el modelo completo con estas fue algo interesante y que faltó para la actividad anterior, las funciones de ambas implementaciones me parecieron interesantes, considero que las 2 actividades se entendieron muy bien. Al final considero que la practica tuvo un buen nivel, al integrar elementos más conocimientos agregar modelos en 3d con un diseño más definido que solo el color que usábamos en los ejercicios y practicas anteriores, estos conceptos agregados en los ejercicios me parecen una aplicación futura en el proyecto final para limitar los movimientos a personajes u objetos.