



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° 06**

**NOMBRE COMPLETO:** Barragán Pilar Diana

**N° de Cuenta:** 318147981

**GRUPO DE LABORATORIO:** 03

**GRUPO DE TEORÍA:** 04

**SEMESTRE 2025-1**

**FECHA DE ENTREGA LÍMITE:** 24 de septiembre de 2024

**CALIFICACIÓN:** \_\_\_\_\_

## Ejercicio de clase práctica 6: Texturizado.

### 1. Texturizar su cubo con la imagen dado\_animales ya optimizada por ustedes.

Para este primer ejercicio modifique la imagen llamada dado\_animales en el programa de GIMP recortando solo el área necesaria, así como escalando la imagen para que cumpla con que mida  $2^n \times 2^n$  en este caso la escale a  $215 \times 215$ , para finalmente exportarla como .tga

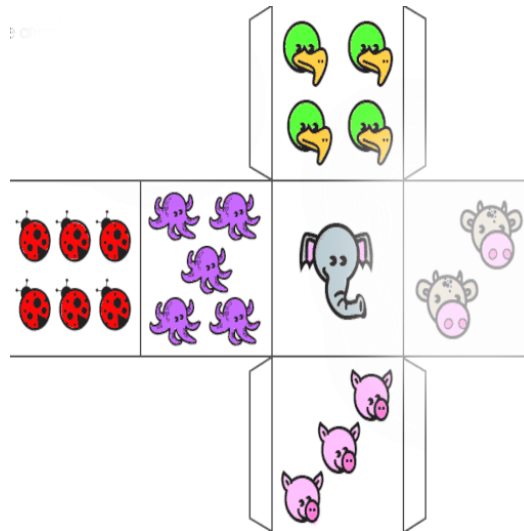


Figura 1. Imagen de GIMP dado\_animales.

Posteriormente texturice en el código el cubo primero agregando el objeto de Texture dadoTexture, para posteriormente en los vértices del cubo colocar las coordenadas de la imagen, ajustando cada una de las caras en la imagen a cada cara del cubo.

```
Texture brickTexture;  
Texture dirtTexture;  
Texture plainTexture;  
Texture pisoTexture;  
Texture dadoTexture;  
Texture logofiTexture;
```

Figura 2. Declaración de la textura dadoTexture.

```

GLfloat cubo_vertices[] = {
    // front
    //x      y      z      S      T      NX      NY      NZ
    -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
    //x      y      z      S      T
    0.5f, -0.5f,  0.5f,  0.51f, 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,  0.51f, 0.66f, -1.0f,  0.0f,  0.0f,
    // back
    -0.5f, -0.5f, -0.5f,  1.0f, 0.34f,  0.0f,  0.0f,  1.0f,
    0.5f, -0.5f, -0.5f,  0.76f, 0.34f,  0.0f,  0.0f,  1.0f,
    0.5f,  0.5f, -0.5f,  0.76f, 0.66f,  0.0f,  0.0f,  1.0f,
    -0.5f,  0.5f, -0.5f,  1.0f, 0.66f,  0.0f,  0.0f,  1.0f,
    // left
    //x      y      z      S      T
    -0.5f, -0.5f, -0.5f,  0.0f, 0.34f,  1.0f,  0.0f,  0.0f,
    -0.5f, -0.5f,  0.5f,  0.24f, 0.34f,  1.0f,  0.0f,  0.0f,
    -0.5f,  0.5f,  0.5f,  0.24f, 0.66f,  1.0f,  0.0f,  0.0f,
    -0.5f,  0.5f, -0.5f,  0.0f, 0.66f,  1.0f,  0.0f,  0.0f,
    // bottom
    //x      y      z      S      T
    -0.5f, -0.5f,  0.5f,  0.51f, 0.0f,  0.0f,  1.0f,  0.0f,
    0.5f, -0.5f,  0.5f,  0.51f, 0.32f,  0.0f,  1.0f,  0.0f,
    0.5f, -0.5f, -0.5f,  0.73f, 0.32f,  0.0f,  1.0f,  0.0f,
    -0.5f, -0.5f, -0.5f,  0.73f, 0.0f,  0.0f,  1.0f,  0.0f,
    //UP
    //x      y      z      S      T
    -0.5f,  0.5f,  0.5f,  0.51f, 1.0f,  0.0f, -1.0f,  0.0f,
    0.5f,  0.5f,  0.5f,  0.51f, 0.67f,  0.0f, -1.0f,  0.0f,
    0.5f,  0.5f, -0.5f,  0.74f, 0.67f,  0.0f, -1.0f,  0.0f,
    -0.5f,  0.5f, -0.5f,  0.74f, 1.0f,  0.0f, -1.0f,  0.0f,
};

```

Figura 3. Coordenadas de la imagen en cada cara del dado.

Finalmente cargue el archivo de la imagen con la textura del cubo en el código, colocando la ruta donde se encuentra el archivo, para poder trasladarlo y escalarlo adecuadamente, así por último cargo la textura y dibujo el cubo.

```

dirtTexture.LoadTextureA();
plainTexture = Texture("Textures/plain.png");
plainTexture.LoadTextureA();
pisoTexture = Texture("Textures/piso.tga");
pisoTexture.LoadTextureA();
//dadoTexture = Texture("Textures/dado-de-numeros.png");
dadoTexture = Texture("Textures/dado_animales.tga");
dadoTexture.LoadTextureA();
logofiTexture = Texture("Textures/escudo_fi_color.tga");
logofiTexture.LoadTextureA();

```

Figura 4. Carga de la textura.

```

color = glm::vec3(1.0f, 1.0f, 1.0f); //color blanco, multiplica a la inf
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, -2.0f, 0.0f));
model = glm::scale(model, glm::vec3(30.0f, 1.0f, 30.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
glUniform3fv(uniformColor, 1, glm::value_ptr(color));

pisoTexture.UseTexture();
meshList[2]->RenderMesh();

```

Figura 5. Dibujado del dado trasladado y escalado

## 2. Importar el cubo texturizado en el programa de modelado con la imagen dado\_animales ya optimizada por ustedes.

Para texturizar el cubo en 3ds Max primero lo cree como objeto con box, posteriormente le agregue la textura del archivo de GIMP, sin embargo, aunque ya se podía ver en el objeto no esta bien acomodado por ello con unwrap UVW edite cada una de las caras del cubo para posicionarlas centradas en su respectivo lugar en las caras del cubo.

Después exporte el cubo en formato .obj modificando en la parte de material -> Map-Export eliminando lo que se encuentra escrito, para finalmente solo crear el objeto Model Dado\_A, cargar el modelo del dado, por último, solo es escalarlo y acomodarlo para cargarlo correctamente en el escenario.

<pre> Texture pisoTexture; Texture dadoTexture; Texture logofiTexture;  Model Kitt_M; Model Llanta_M; Model Dado_M; Model Dado_A; </pre>	<pre> Kitt_M = Model(); Kitt_M.LoadModel("Models/kitt_optimizado.obj"); Llanta_M = Model(); Llanta_M.LoadModel("Models/llanta_optimizada.obj");  Dado_A = Model(); Dado_A.LoadModel("Models/dado_animales.obj"); </pre>
--	---

Figura 6. Declaración y carga del objeto Model Dado\_A.

```
//Ejercicio 2: Importar el cubo texturizado en el programa de modelado con
//la imagen dado_animales ya optimizada por ustedes
//Dado importado
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(-5.0f, 4.5f, -2.0f));
model = glm::scale(model, glm::vec3(0.05f, 0.05f, 0.05f));
model = glm::rotate(model, 180 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Dado_A.RenderModel();
```

Figura 7. Posicionamiento y dibujo del dado a partir de un modelo.

## EJECUCIÓN DEL PROGRAMA:

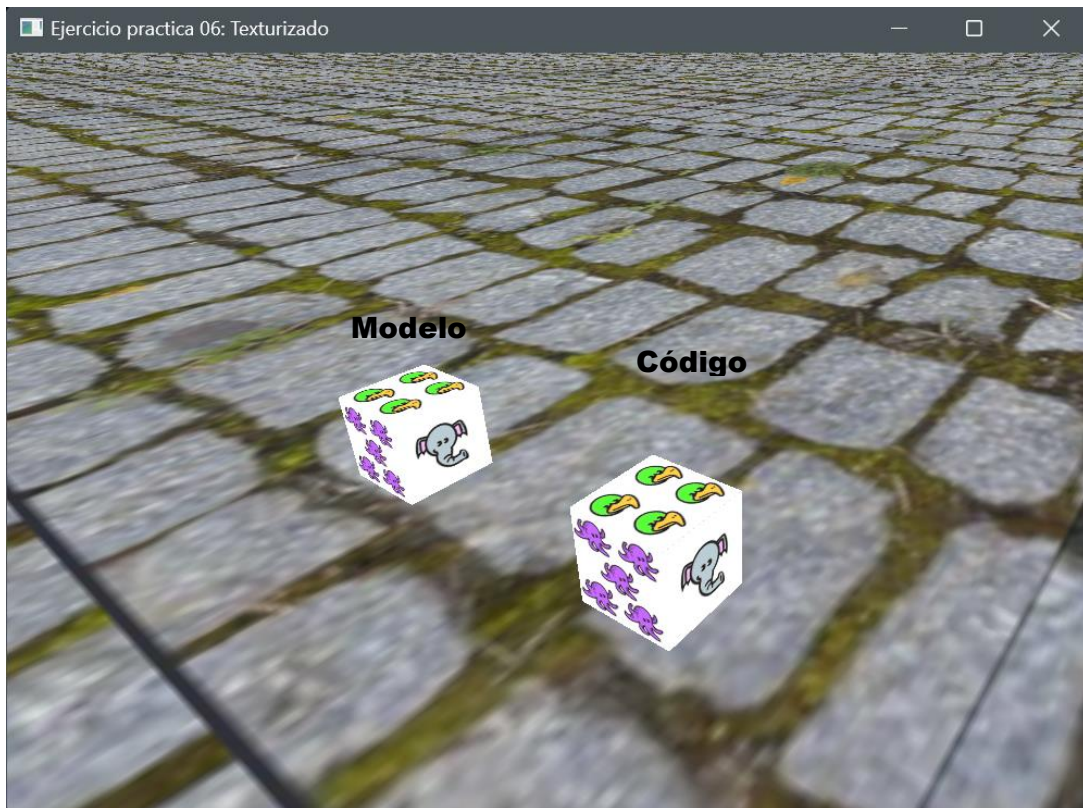


Figura 8. Dado de animales.

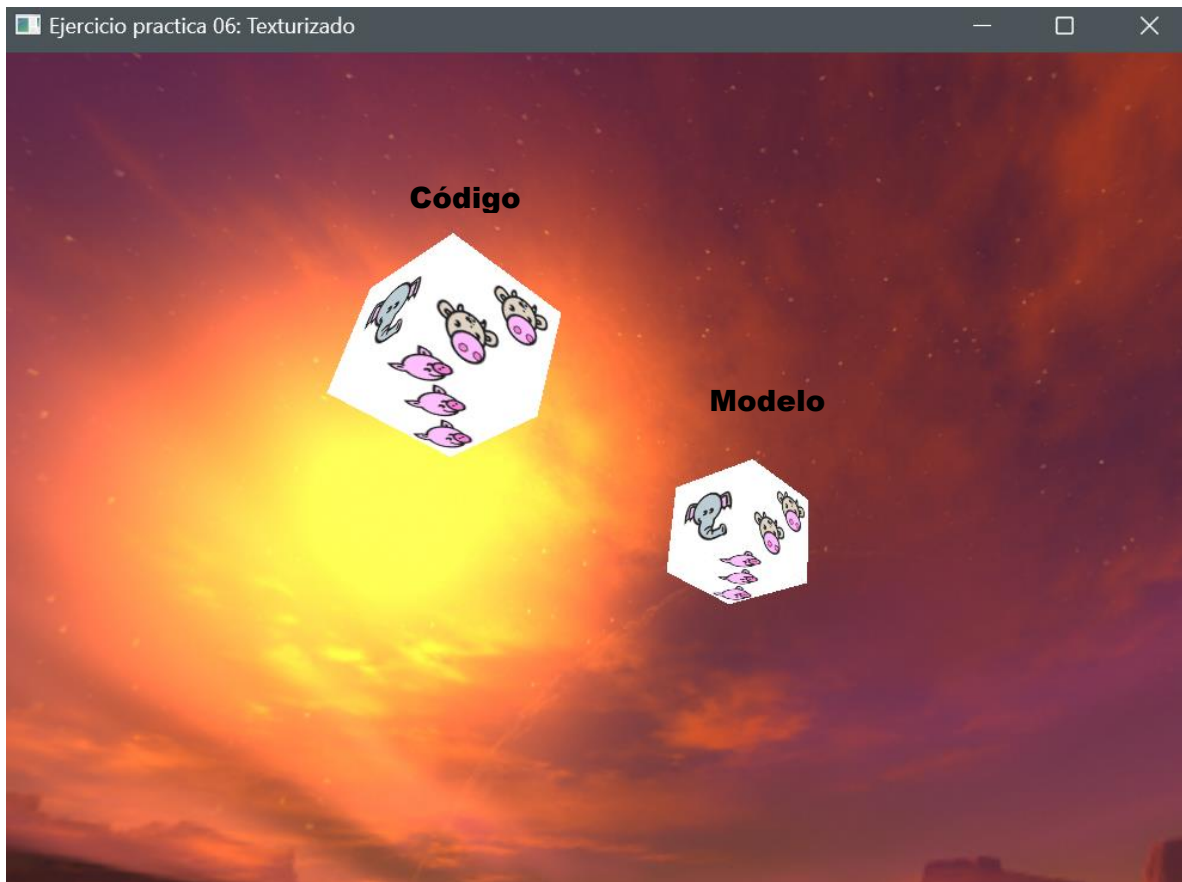


Figura 9. Dado de animales.

En este ejercicio no tuve inconveniente alguno a la hora de compilar y ejecutar el código.

### **CONCLUSIONES:**

Con la realización del ejercicio me fue posible aprender a utilizar un conjunto variado de programas puesto que para la edición de las imágenes se utilizó GIMP y código en opengl para agregarle la textura al dado, el mismo resultando lo obtuve pero ahora haciendo uso de 3ds Max conociendo nuevas herramientas para poder darle el texturizado y no solo eso sino acomodarlo de manera óptima.