



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° 9-2**

**NOMBRE COMPLETO:** Barragán Pilar Diana

**N° de Cuenta:** 318147981

**GRUPO DE LABORATORIO:** 03

**GRUPO DE TEORÍA:** 04

**SEMESTRE 2025-1**

**FECHA DE ENTREGA LÍMITE:** 22 de octubre de 2024

**CALIFICACIÓN:** \_\_\_\_\_

## Práctica 9-2: Animación Avanzada.

### 1. Alternar entre la textura del número 1 y 2.

Para lograr esto utilice la función `glfwGetTime()` que devuelve en tiempo en segundos desde que se inicio la ejecución del programa y va aumentando conforme este se ejecuta, también utilice `fmod()` esto para obtener el modulo del valor que devuelve `glfwGetTime()` entre en este caso el valor de 2 para que dado un número par el resto sea de 1 y para un impar el resto sea de cero, entonces con un `if` si el resultado es menor que uno se pondrá la textura del número 1, mientras que si el resto que nos de es de 1 con los números pares la textura que se pondrá será del número dos.

```
double alter = fmod(glfwGetTime(), 2);
//printf("\nalter %f \n", alter);

if (alter < 1) {
    Numero2Texture.UseTexture();
}
else {
    Numero1Texture.UseTexture();
}
```

Figura 1. Modulo para alternar e if para uso de la textura.

### 2. Agregar que el número cambiante sea a una velocidad visible.

Primero declare una variable fuera del `while` de tipo `int` llamada `t`, esto para que donde se encuentra el código del número cambiante se modifique la línea `toffsetnumerocambiau += 0.25` con un `if` que la ejecute cada determinado tiempo que dicta la variable `t` que inicialice en 3, y posteriormente se vaya aumentando el tamaño de la variable para que se mantenga la velocidad y sea posible ver el cambio.

```

//número cambiante
/*
¿Cómo hacer para que sea a una velocidad visible?
*/

if (glfwGetTime() > t) {
    toffsetnumerocambiau += 0.25;
    t = 1 + t;
}

if (toffsetnumerocambiau > 1.0)
    toffsetnumerocambiau = 0.0;
toffsetnumerov = 0.0;
toffset = glm::vec2(toffsetnumerocambiau, toffsetnumerov);
model = glm::mat4(1.0);

```

Figura 2. if para que el numero cambie a velocidad visible.

### 3. Separar alas del dragón.

Como en anteriores prácticas separe las alas del cuerpo del modelo además de centrar cada una de estas partes en 3ds Max y las exporte para posteriormente utilizarlas en el programa de OpenGL declarando los modelos, especificando las rutas donde se encuentran y finalmente posicionándolos adecuadamente con traslate.

```

//HIDRA
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, 5.0f+sin(glm::radians(angulovaria)), 6.0));
//model = glm::translate(model, glm::vec3(0.0f, 5.0f, 6.0));
model = glm::scale(model, glm::vec3(0.6f, 0.6f, 0.6f));
modelaux = model;
model = glm::rotate(model, -90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
/*color = glm::vec3(0.0f, 1.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));*/
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Hidra.RenderModel();

//Ala Derecha
model = modelaux;
model = glm::translate(model, glm::vec3(-0.4f, 2.9f, 0.43f));
model = glm::rotate(model, -90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
AlaD.RenderModel();

//Ala Izquierda
model = modelaux;
model = glm::translate(model, glm::vec3(-0.63f, 2.9f, -0.63f));
model = glm::rotate(model, -90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
AlaI.RenderModel();

```

Figura 3. Posicionamiento del modelo.

## **CONCLUSIONES:**

Con este ejercicio puede comprender de mejor manera como funciona el movimiento en OpenGL ya no solo de modelos sino también de texturas, al igual que la animación utilizando ciertos tipos de formas como lo fue la onda senoidal para el modelo de la hidra, modificando el código para entenderlo que mejor manera e implementando funciones sumamente útiles como el intercambiar una textura.

## **REFERENCIAS:**

IBM. (2024, 07 de octubre). fmod () - Calcular resto de coma flotante. Recuperado el 22 de octubre de 2024, de: <https://www.ibm.com/docs/es/i/7.5?topic=functions-fmod-calculate-floating-point-remainder>