



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

NOMBRE COMPLETO: Araiza Valdés Diego Antonio

N° de Cuenta: 423032833

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 06

SEMESTRE: 2026-1

FECHA DE ENTREGA LÍMITE: 28/10/2026

CALIFICACIÓN: _____

EJERCICIOS DE SESIÓN:

1. Actividades realizadas. Una descripción de los ejercicios y capturas de pantalla de bloques de código generados y de ejecución del programa

1.1. Actividad 1: Agregar que el número cambiante sea a una velocidad visible.

```
float cambia = 0.0f;
float toffsetvelocidadnum = 0.0f;
bool num2cambia = true;
```

Primero se declararon las variables y banderas que utilizaremos.

```
//animacion cambio numero
if (cambia > 1.0f) {
    toffsetvelocidadnum = 1.0f;
    cambia = 0.0f;
}
else {
    toffsetvelocidadnum = 0.0f;
    cambia += 0.025 * deltaTime;
}
```

Luego se hizo uso de un if para reducir en 40 el deltaTime y de esta manera hacer más lento el cambio de números

```
toffsetnumerocambiau += 0.25 * toffsetvelocidadnum;
if (toffsetnumerocambiau > 0.75)
    toffsetnumerocambiau = 0.0;
toffsetnumerov = 0.0;
```

Finalmente se multiplicó por la bandera toffsetvelocidadnum para que solo se recorriera las 0.25 unidades cuando este valiera 1.

1.2. Actividad 2: Alternar entre la textura del número 1 y 2.

```
if (toffsetvelocidadnum >= 1) {
    num2cambia = !num2cambia;
}

if (num2cambia) {
    Numero2Texture.UseTexture();
}
else {
    Numero1Texture.UseTexture();
}
```

Para este ejercicio se utilizó el mismo intervalo que en el anterior, por lo que con cada intervalo se cambio el valor de un booleano y a partir de ahí se cambió entre una textura y otra.

1.3.Actividad 3: Separarle las alas al dragón, hacer que el dragón avance 20 unidades de forma senoidal aleteando y regrese volteado/ no de espaldas (loop)



Primero se separó las alas del modelo en Blender y se exportaron cada una por aparte.

```
bool aletea = true;
float rotalas;
float rotalasOffset;
bool dragonavanza = true;
int rotdragon = 0;
float posdragonxz = 0.0f;
rotalasOffset = 2.5f;
posdragonxz = 1.0f;
```

Posteriormente se crearon e inicializaron las variables y banderas que utilizamos para mover, rotar y animar el dragón

```
if (aletea)
{
    if (rotalas > -90.0f)
    {
        rotalas -= rotalasOffset * deltaTime;
    }
    else
    {
        aletea = !aletea;
    }
}
else
{
    if (rotalas < 0.0f)
    {
        rotalas += rotalasOffset * deltaTime;
    }
    else
    {
        aletea = !aletea;
    }
}
```

Posteriormente se hizo uso de un if con el booleano aletea, de modo que aletee hacia abajo haciendo uso de deltaTime, y cuando deje de aletear (llegó al punto mínimo) cambie el sentido de la rotación.

```
if (dragonavanza)
{
    if (dragonavance > -20.0f)
    {
        dragonavance -= avanceoffset * deltaTime;
    }
    else
    {
        dragonavanza = !dragonavanza;
        rotdragon = 180;
        posdragonxz = -1.0f;
    }
}
else
{
    if (dragonavance < 20.0f)
    {
        dragonavance += avanceoffset * deltaTime;
    }
    else
    {
        dragonavanza = !dragonavanza;
        rotdragon = 0;
        posdragonxz = 1.0f;
    }
}
```

Por consiguiente, se hizo uso de otro if, esta vez para el movimiento, de manera el dragón avance hacia enfrente, y cuando deje de avanzar (llegó al punto máximo) cambie el valor de la bandera, rote el dragón e invierta el valor de posdragón para acomodar de manera correcta las alas.

```
//dragonavance para moverlo hacia adelante y atrás en X
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f + dragonavance, 5.0f + 3 * sin(glm::radians(angulovaria)), 6.0));
modelaux = model;
model = glm::scale(model, glm::vec3(0.3f, 0.3f, 0.3f));
model = glm::rotate(model, rotdragon * 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));
Dragon_M.RenderModel();

//Ala izquierda
model = modelaux;
model = glm::translate(model, glm::vec3(0.544f*posdragonxz, 1.071f, 0.145f * posdragonxz));
model = glm::scale(model, glm::vec3(0.3f, 0.3f, 0.3f));
model = glm::rotate(model, rotdragon * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::rotate(model, glm::radians(rotalas), glm::vec3(-0.8910f, 0.4540f, 0.0f));
Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
DragonAI_M.RenderModel();
```

```

//Ala derecha
model = modelaux;
model = glm::translate(model, glm::vec3(0.544f * posdragonxz, 1.071f, -0.033 * posdragonxz));
model = glm::scale(model, glm::vec3(0.3f, 0.3f, 0.3f));
model = glm::rotate(model, rotdragon * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::rotate(model, glm::radians(rotalas), glm::vec3(0.8910f, -0.4540f, 0.0f));
Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
DragonAD_M.RenderModel();

```

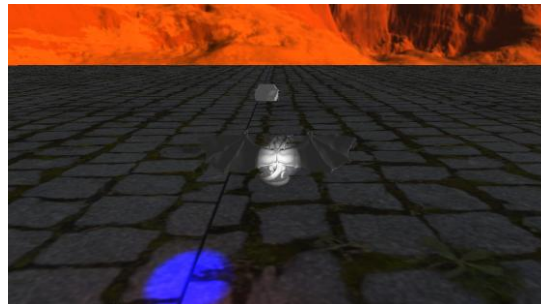
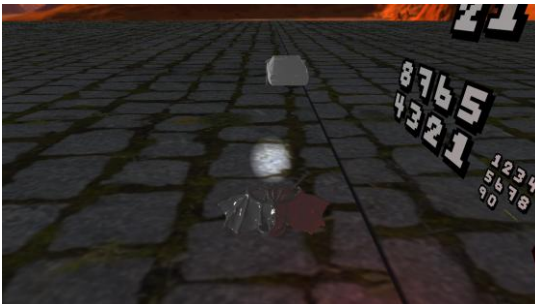
Posteriormente se declaró el dragón y las alas, donde se les asignó la animación de traslación senoidal, la rotación rotdragon al dragón y alas a las alas para que giren al llegar al punto máximo, se multiplicó por posdragonxz las componentes X y Z de las alas para que se acomodan al rotar 180 grados y se les aplicó la rotación rotalas para que animarlas.

1.4. Evidencias:

1.4.1. Ejercicio 1 y 2:



1.4.2. Ejercicio 3:



2. Problemas presentados. Listar si surgieron problemas a la hora de ejecutar el código

R: No hubo ningún problema

3. Conclusión:

- 3.1. Los ejercicios de la clase: Complejidad, explicación

El ejercicio fue bastante adecuado, no hubo mayor complicación para su realización, bastante adecuado.

- 3.2. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias.

Respecto a la explicación fue buena, explicó a detalle lo esencial, sobre todo lo que es animación y lo que no es y el if para el movimiento del auto fue de gran ayuda para realizar el ejercicio.