



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° 09

NOMBRE COMPLETO: Aguilar Pérez José Ramón

N° de Cuenta: 317515048

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 23/abril/2025

CALIFICACIÓN: _____

EJERCICIOS DE SESIÓN:

1. Actividades realizadas.

- 1.- Agregar que el número cambiante sea a una velocidad visible.
- 2.- Alternar entre la textura del número 1 y 2
- 3.- Separar alas del dragón y que aletee.

Para el primer ejercicio, se agregó un temporizador para que el cambio de número sea visible. Este temporizador utiliza la variable *deltaTime* para que cada medio segundo se haga el cambio de número, además de que se reinicia el temporizador para que el ciclo de los números se mantenga constante. Al aplicar estos cambios, se pudo apreciar que los números de la secuencia son del 1 al 4.

```
//Cambio visible de numeros
float Temporizador = 0.0f;

Temporizador += deltaTime * 0.025f;
if (Temporizador >= 0.5f) { // cambia cada 0.5 segundos
    Temporizador = 0.0f; // reinicia el tiempo

    toffsetnumerocambiau += 0.25f;
    if (toffsetnumerocambiau > 0.75f) { //Se pone a 0.75 para que no se repita el 1 en el ciclo
        toffsetnumerocambiau = 0.0f;
    }
}
```

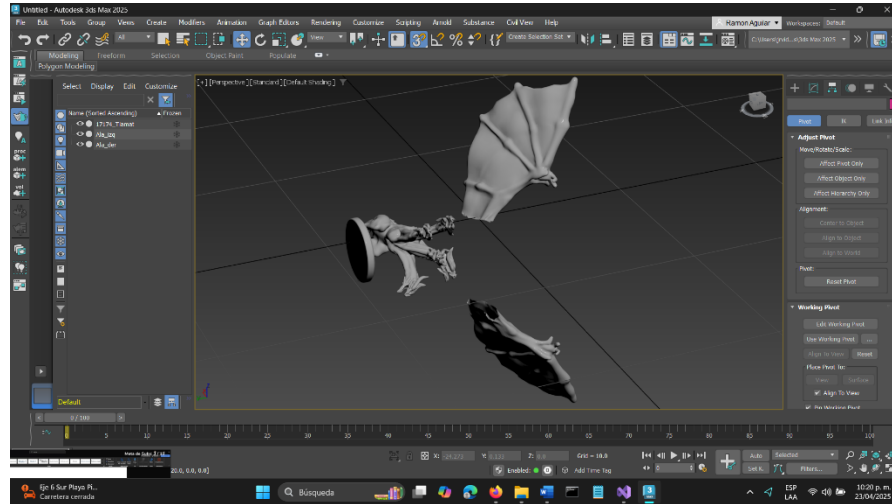
Para el segundo ejercicio, se hizo algo similar. Se estableció otro temporizador, esto con el fin de evitar conflictos a la hora de ejecutar, y se agregó una bandera booleana *textNum*, la cual alterna entre true y false cada que el temporizador se reinicia. Después, en la condicional se evalúa la bandera *textNum*, si es verdadera mostrará la textura del número 1 y si es falsa la del 2.

```
//Cambio de texturas 1 y 2
float TempoText = 0.0f;
bool textNum = true;

//cambiar automáticamente entre textura número 1 y número 2
TempoText += deltaTime * 0.025f;
if (TempoText >= 1.0f) {
    TempoText = 0.0f; // reinicia el tiempo
    textNum = !textNum; //Alternar entre true y false
}

if (textNum) {
    Numero1Texture.UseTexture(); //textNum -> True
}
else {
    Numero2Texture.UseTexture(); //textNum -> False
}
```

Para el último ejercicio, se separaron las alas del dragón, las cuales se les acomodó el pivote para que su movimiento fuera el correcto. Cada una de las partes del dragón se exportaron al proyecto del ejercicio. Como en ejercicios y prácticas pasadas se declararon los modelos en el programa y se instanció el dragón por medio de jerarquía utilizando matrices auxiliares.



Para el movimiento de las alas, se crearon dos variables similares a las utilizadas para el movimiento de las llantas del auto. Antes del ciclo while de la ventana del programa, se inicializan las variables: *rotAla* es la que llevara el control del angulo máximo y mínimo al que llegan las alas, mientras que *rotAlaOffset* controla la velocidad del giro de las alas.

```
float rotAla;
float rotAlaOffset;
rotAla = 0.0f;
rotAlaOffset = 1.0f;
```

Dentro del while de la ventana del programa, se instancia la condicional de las alas. Al llegar a -30° (que sería el movimiento hacia arriba del ala) deja de girar el ala y empieza su movimiento en sentido opuesto, lo mismo pasa al llegar a 50° , el ala cambia de sentido de giro.

```
//Animación simple de las alas
rotAla += rotAlaOffset * deltaTime;
// Cambia de dirección al alcanzar los límites
if (rotAla > 50.0f) {
    rotAla = 50.0f; //Limite para evitar atasco
    rotAlaOffset *= -1.0f;
}
else if (rotAla < -30.0f) {
    rotAla = -30.0f; //Limite para evitar atasco
    rotAlaOffset *= -1.0f;
}
```

Finalmente, se agrega este movimiento, *rotAla*, a las rotaciones de las alas, las cuales giran en torno al eje X. Además, se agregó el movimiento senoidal en la traslación del cuerpo del dragón para que así empiece su animación de “avanzar”.

```
//Instancia del dragon, angulovaria -> angulo sinoidal
model = glm::mat4(1.0f);
//model = glm::translate(model, glm::vec3(0.0f, 5.0f+sin(glm::radians(angulovaria)), 6.0f));
model = glm::translate(model, glm::vec3(0.0f-angulovaria*0.01, 5.0f + 2.0*sin(glm::radians(angulovaria)), 6.0f));
model = glm::scale(model, glm::vec3(0.3f, 0.3f, 0.3f));
model = glm::rotate(model, -90 * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
modelaux = model;
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));
Tiamat_M.RenderModel();

//Ala izquierda del dragon
model = modelaux;
model = glm::translate(model, glm::vec3(-1.0f, 0.0f, 3.0f));
model = glm::rotate(model, rotAla * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Ala_Izq_M.RenderModel();

//Ala derecha del dragon
model = modelaux;
model = glm::translate(model, glm::vec3(-1.0f, 0.0f, 3.0f));
model = glm::rotate(model, -rotAla * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Ala_Der_M.RenderModel();
```



Da clic en la imagen para ver el video del funcionamiento del programa :D

2. Problemas presentados.

No se presentaron problemas a la hora de realizar el ejercicio.

3. Conclusión

a. Los ejercicios de la clase: Complejidad, explicación

Los ejercicios solicitados tuvieron una complejidad interesante, ya que los distintos tipos de animación se pueden abarcar desde varias perspectivas.

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

La explicación general de los distintos tipos de animación, así como el funcionamiento de cada uno de estos fue clara y precisa para la elaboración de las actividades solicitadas para este ejercicio. Es interesante como se pueden abordar desde distintos puntos de vista los diferentes problemas planteados.