



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

## MANUAL TÉCNICO

Grupo: 04

Profesor:


Román Balbuena Carlos Aldair

Alumno:

Mora González Alan Francisco


Semestre 2022-2

19 de mayo de 2022

	<b>Manual técnico Proyecto Final</b>	No. Pagina:	1
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería		Computación grafica e interacción humano computadora	

## Índice general

<b>Objetivo .....</b>	<b>2</b>
<b>Descripción .....</b>	<b>2</b>
<b>Alcances .....</b>	<b>2</b>
<b>Cronograma de actividades .....</b>	<b>2</b>
<b>Imágenes de referencia .....</b>	<b>4</b>
<b>Código .....</b>	<b>6</b>
Libraries .....	6
External libraries .....	6
Include files .....	7
Shaders .....	7
Funciones .....	7
Variables globales .....	8
Carga de modelos .....	9
Animaciones .....	10
<b>Análisis de costos .....</b>	<b>12</b>
<b>Conclusiones .....</b>	<b>13</b>

	<b>Manual técnico Proyecto Final</b>	No. Pagina:	2
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería		Computación grafica e interacción humano computadora	

## Objetivo

El alumno deberá aplicar y demostrar los conocimientos adquiridos durante todo el curso.

## Descripción

El alumno deberá seleccionar una fachada y un espacio que pueden ser reales o ficticios y presentar imágenes de referencia de dichos espacios para su recreación 3D en OpenGL.

En la imagen de referencia se debe visualizar 5 objetos que el alumno va a recrear virtualmente y donde dichos objetos deben ser lo más parecido a su imagen de referencia, así como su ambientación.


## Alcances

Se busca recrear el espacio animado: *McDonald's* centrándose en la fachada exterior del restaurante, así como la sala de estar junto con parte de la cocina, además de diseñar un baño para el restaurante en OpenGL utilizando el modelado 3d para realizar los objetos del ambiente a realizar con su respectivo texturizado, esto mediante el software Maya Autodesk para posteriormente cargar dichos modelos 3D en OpenGL utilizando modelado jerárquico y transformaciones geométricas. Además, aplicando los conocimientos del pipeline gráfico como son shaders, cámara, dibujado, proyección, texturizado, iluminación, etc.

Dicho espacio tendrá interacción con el usuario mediante el uso de la cámara para recorrerlo, así como animaciones sencillas y complejas que ofrezcan una experiencia sencilla de inmersión en el contexto para el mismo.

## Cronograma de actividades

Actividades	Inicio	Duración (días)	Fin
Planificación	19 de abril	5	24 de abril
Búsqueda de referencias	19 de abril	4	23 de abril
Creación del proyecto	24 de abril	1	25 de abril

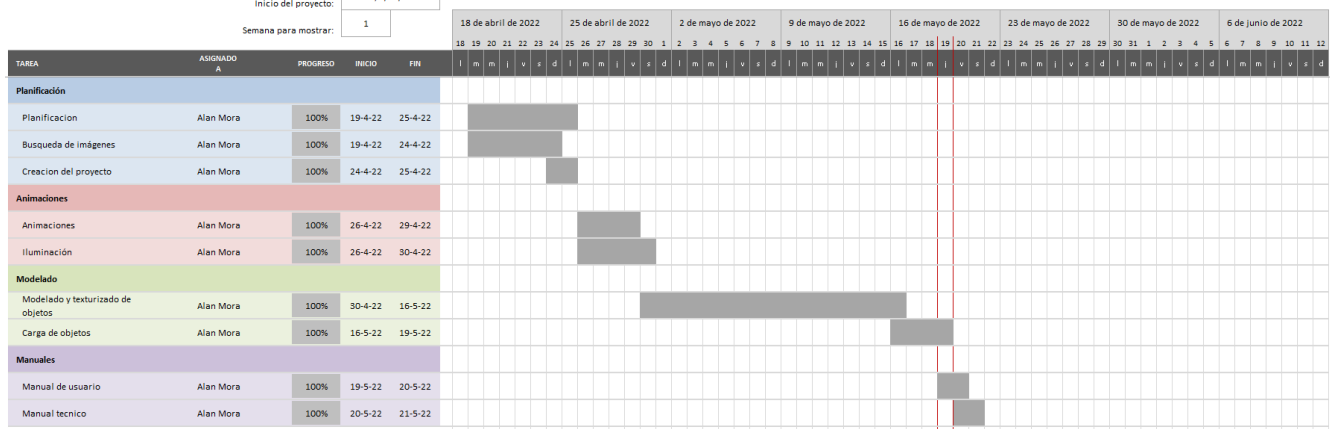
	<b>Manual técnico Proyecto Final</b>	No. Pagina:	3
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería		Computación grafica e interacción humano computadora	


Animaciones	26 de abril	4	29 de abril
Iluminación	26 de abril	5	30 de abril
Modelado y texturizado de objetos	30 de abril	16	16 de mayo
Carga de objetos	16 de mayo	3	19 de mayo
Manual de usuario	19 de mayo	1	20 de mayo
Manual técnico	20 de mayo	1	21 de mayo

## PROYECTO FINAL

Orange Corp  
Alan Francisco Mora


Inicio del proyecto: mar, 4/19/2022  
Semana para mostrar: 1



	<b>Manual técnico</b> <b>Proyecto Final</b>	No. Pagina:	4
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería		Computación grafica e interacción humano computadora	

## Imágenes de referencia



	<b>Manual técnico Proyecto Final</b>	No. Pagina:	5
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería		Computación grafica e interacción humano computadora	


Objetos a recrear:

- Mesa
- Silla
- Sillón
- Máquina de auto cobro
- Lámparas
- Bote de basura
- Barra de pedidos



Objetos a recrear:

- Lavabo
- Dispensador de jabón
- Tasa de baño
- Secador de manos
- Barra de lavabos

	<b>Manual técnico</b> <b>Proyecto Final</b>	No. Pagina:	6
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería		Computación grafica e interacción humano computadora	

## Código

### Libraries

```
// Std Includes
#include <string>
#include <iostream>
#include <cmath>

// GLEW
#include <GL/glew.h>

// GLFW
#include <GLFW/glfw3.h>

// GL includes
#include "Shader.h"
#include "Camera.h"
#include "Model.h"


// GLM Mathemtics
#include <glm/glm.hpp>
#include <glm/gtc/matrix_transform.hpp>
#include <glm/gtc/type_ptr.hpp>

// Other Libs
#include "SOIL2/SOIL2.h"
#include "stb_image.h"
```

### External libraries

Biblioteca	Descripción
GLEW, GLFW	Para la comunicación con el SO y para el manejo de ventanas en Visual Studio y OpenGL (creación, eliminación, etc.).
GLM	Biblioteca matemática de C++ para proyectos sobre la API OpenGL.
Assimp	Biblioteca para importar modelos 3d de programas como 3ds MAX, Maya, Blender, etc. y que OpenGL pueda entender su formato, en nuestro caso permite importar modelos .obj y leer los archivos .mtl con la información de estos objetos para abstraerse a OpenGL.
SOIL2	Permite cargar texturas (imágenes) en OpenGL.



	<b>Manual técnico Proyecto Final</b>	No. Pagina:	7
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería		Computación grafica e interacción humano computadora	

## Include files

Biblioteca	Descripción
Shader.h	Crea el vertex shader y fragment shader, lo único que recibe son dos direcciones, la primera para el archivo de vertex shader y la segunda con el archivo del fragment shader. Abre dichos archivos recuperando el código que contienen y posteriormente crea ambos shader asignándoles localidades de memoria, se compilan, verifica que se hayan creado correctamente.
Camera.h	Crea la cámara sintética o interactiva, este .h define los movimientos que tendrá la cámara (adelante, atrás, izquierda y derecha), las posiciones o valores iniciales de la cámara para la rotación, inclinación, velocidad de movimiento, sensibilidad y zoom o ángulo de apertura para la proyección en perspectiva.
Model.h	Recibe la dirección del archivo .obj y se encarga de importar dicho modelo utilizando assimp, cargando coordenadas de sus vértices (posiciones, normales y de texturas).


## Shaders

Shaders	Descripción
lamp.vs y lamp.frag	El vertex accede a la localidad de memoria del VAO con la información sobre los vértices de posición y se afecta a las matrices con la posición, el fragment despliega la información de color.
lighting.vs y lighting.frag	El vertex shader accede a todas las coordenadas del VAO (posición, color y textura), recibe las matrices de modelo, vista y proyección para modificarlas según la geometría recibida del VAO. El fragment shader define los tipos de luz que estarán disponibles (directional, point y spot) en forma de estructura con sus componentes.

## Funciones

```
// Function prototypes
void KeyCallback(GLFWwindow *window, int key, int scancode, int action, int mode);
void MouseCallback(GLFWwindow *window, double xPos, double yPos);
void DoMovement();
void animacionChairs();
void animacionBurger();
```




	<b>Manual técnico Proyecto Final</b>	No. Pagina:	8
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería		Computación grafica e interacción humano computadora	

Shaders	Descripción
KeyCallback	Procesa la entrada por teclado, las teclas presionadas para realizar una acción. Se consideran 5 teclas en específico para la interacción con el ambiente virtual [ESC, W, A, S, D]. También se consideran los eventos de las teclas 0 a 1023.
MouseCallback	Es el listener de la entrada por mouse. La acción que define para dicha entrada es la actualización en la rotación de la cámara.
DoMovement	Esta función define las acciones a realizar si hay animaciones activadas iniciadas en KeyCallback, cuando cambian de estado y también la actualización de la cámara.
animacionChairs	Dentro de esta animación esta la programación necesaria para hacer la animación compleja de las sillas del restaurante.
animacionBurger	Dentro de esta animación esta la programación necesaria para hacer la animación compleja del tiro parabólico utilizando una hamburguesa restaurante.
main	Función principal con el programa para establecer el ambiente virtual. Crea la ventana, verifica errores, llama a las funciones "listener" o callbacks para los eventos de teclado y mouse. Define el viewport, activa opciones de OpenGL para el manejo de materiales translúcidos y/o transparentes. Crea y configura los shaders. Captura los eventos de los listeners de movimiento y de entradas de teclado y mouse. Define la iluminación.

### Variables globales

Variable	Descripción
camera	Tipo Camera con la posición inicial de la cámara en el espacio.
keys	Arreglo de booleanos para manejar la entrada por teclado.
firstMouse	Booleano para el manejo de la cámara.
WIDTH	Variable con el ancho de la ventana.
HEIGHT	Variable con el alto de la ventana.
SCREEN_WIDTH y SCREEN_HEIGHT	Para almacenar el ancho y alto que devuelva alguna función.
transChair, rotChair y activeChair	Variables para la animación sencilla de la silla.
transTrash, rotTrash y activeTras	Variables para la animación sencilla del bote de basura.
firstMouse	Booleano para el manejo de la cámara.
movKitXChair, movKitYChair y rotKitChair	Variables para la animación compleja de las sillas.
circuitoChair (1, 2)	Variables booleanas para activar y desactivar la animación compleja de las sillas, dependiendo de estas variables y el recorrido que se activa será la posición de las sillas en ese momento.

	<b>Manual técnico Proyecto Final</b>	No. Pagina:	9
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería		Computación grafica e interacción humano computadora	

vinY, vinZ y rotKitBur	Variables para la animación compleja de la hamburguesa.
circuitoBur (1, 2)	Variables booleanas para activar y desactivar la animación compleja de la hamburguesa, dependiendo de estas variables y el recorrido que se activa será la posición de la hamburguesa en ese momento.
recorrido (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	Variables booleanas para los estados de las animaciones complejas definidas en KeyCallback.
lightPos	Atributo posición de la iluminación.
pointLightPositions	Arreglo con las coordenadas de 5 point lights.
deltaTime	Tiempo entre el fotograma actual y el último fotograma.
lastFrame	Tiempo del último fotograma.

### Carga de modelos (Objetos).


Los modelos son cargados a través de la creación de un Objeto Model para cada modelo. Utilizando Model.h especificando para cada modelo la ruta del archivo .obj para que la biblioteca o plantilla se encargue de cargar los modelos como se explica en la sección de include files.

```
// Modelos
Model chair ((char*) "Models/Silla/silla.obj");
Model table ((char*) "Models/Table/table.obj");
Model sofa ((char*) "Models/Sofa/sofa.obj");
Model mostrador ((char*) "Models/Mostrador/mostrador.obj");
Model trash ((char*) "Models/Trash/trash.obj");
Model tapa ((char*) "Models/Trash/tapa.obj");
Model maquina ((char*) "Models/Maquina/maquina.obj");
Model lamp ((char*) "Models/Lamp/lamp.obj");
Model burger ((char*) "Models/Burger/burger.obj");
Model charola ((char*) "Models/Burger/charola.obj");

// Interior
Model interior ((char*) "Models/Interior/interior.obj");
Model pilar ((char*) "Models/Interior/pilar.obj");

// Exterior
Model left ((char*) "Models/Restaurante/left.obj");
Model right ((char*) "Models/Restaurante/right.obj");
Model techo ((char*) "Models/Restaurante/techo.obj");

// Bathroom
Model toilet ((char*) "Models/Toilet/toilet.obj");
Model lavabos ((char*) "Models/Lavabos/lavabos.obj");
Model secador ((char*) "Models/Secador/secador.obj");
Model espejos ((char*) "Models/Espejos/espejos.obj");
```

	<b>Manual técnico Proyecto Final</b>	No. Pagina:	10
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería	Computación grafica e interacción humano computadora		

```
Model jabonera((char*)"Models/Jabonera/jabonera.obj");
```


## Animaciones

La activación de las animaciones las encontramos en la función de KeyCallback:

```
if (keys[GLFW_KEY_R])
{
    activeChair = !activeChair;
    if (activeChair)
    {
        transChair -= 1.0f;
        rotChair += 50.0f;
    }
    else
    {
        transChair += 1.0f;
        rotChair -= 50.0f;
    }
}

if (keys[GLFW_KEY_F])
{
    activeTrash = !activeTrash;
    if (activeTrash)
    {
        rotTrash += 60.0f;
        transTrash += 2.2f;
    }
    else
    {
        rotTrash -= 60.0f;
        transTrash -= 2.2f;
    }
}

if (keys[GLFW_KEY_T])
{
    active = !active;
    if (active)
    {
        Light1 = glm::vec3(0.0f, 0.0f, 1.0f);
        Light2 = glm::vec3(0.0f, 0.0f, 1.0f);
        Light3 = glm::vec3(0.0f, 0.0f, 1.0f);
    }
}
```

	<b>Manual técnico</b> <b>Proyecto Final</b>	No. Pagina:	11
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería		Computación grafica e interacción humano computadora	


```

        Light4 = glm::vec3(0.0f, 0.0f, 1.0f);
        Light5 = glm::vec3(0.0f, 0.0f, 1.0f);
    }
    else
    {
        Light1 = glm::vec3(0);
        Light2 = glm::vec3(0);
        Light3 = glm::vec3(0);
        Light4 = glm::vec3(0);
        Light5 = glm::vec3(0);
    }
}

if (keys[GLFW_KEY_Z])
{
    if (circuitoChair1)
    {
        circuitoChair1 = false;
        circuitoChair2 = true;
    }
    else
    {
        circuitoChair1 = true;
        circuitoChair2 = false;
    }
}

if (keys[GLFW_KEY_X])
{
    if (circuitoBur1)
    {
        circuitoBur1 = false;
        circuitoBur2 = true;
    }
    else
    {
        circuitoBur1 = true;
        circuitoBur2 = false;
    }
}

```

	<b>Manual técnico</b> <b>Proyecto Final</b>	No. Pagina:	12
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería		Computación grafica e interacción humano computadora	

Las primeras tres animaciones consisten en rotar y trasladar objetos, se animaron los botes de basura y las sillas del restaurante. Además, se hizo como animación sencilla un encendido y apagado de luces.

Las otras dos animaciones son complejas, el movimiento de sillas hace transformaciones un poco más complejas que las sencillas, al medir las rotaciones y traslaciones necesarias para simular el levantamiento de las sillas. Mientras que el salto que hace la hamburguesa hace uso del tiro parabólico.

## Análisis de costos

Dentro del proyecto se contemplan los sueldos de diversas personas que se harán cargo de tareas según su puesto dentro de la empresa. Estos se desglosan en la siguiente tabla (considerando que un día de trabajo tiene 7 horas).

Puesto	Sueldo por día	Horas de trabajo	Días de trabajo	Total
Líder de proyecto	\$1060	42	6	\$6360
Líder comercial	\$950	21	3	\$2850
Diseñador	\$800	112	16	\$12800
Programador	\$700	91	13	\$9100
Total				<b>\$31100</b>

### Recursos humanos

Considerando holgura del 10% y una utilidad del 60%

$$RH = (1.6)(31100) + (0.1)(31100)$$


$$RH = \$52870$$

### Servicios

- Internet.
  - \$700 al mes
- Energía eléctrica.
  - \$150 promedios mensuales

$$Servicios = 700 + 150$$

$$Servicios = \$850$$

	<b>Manual técnico Proyecto Final</b>	No. Pagina:	13
		Fecha de emisión:	19 de mayo de 2022
Facultad de Ingeniería		Computación grafica e interacción humano computadora	

## Herramientas de apoyo

- Papelería.
  - \$150 mensuales.

$$Papeleria = \$150$$

## Pago neto (PN)

$$PN = RH + Servicios + Papeleria$$

$$PN = 52870 + 850 + 150$$

$$PN = \$53870$$

Por lo que el precio total del proyecto es de \$53,870.00 con cuatro semanas de trabajo equivalentes a un mes, únicamente teniendo lo acordado en las imágenes de referencia. Se deberá pagar una parte proporcional del 30% para comenzar con el mismo y liberar el total del proyecto al finalizar el mes, junto con la entrega.

## Conclusiones

Durante este proyecto pude aplicar los conceptos vistos en clase durante todo el curso, comprendí sobre el modelado y texturizado de objetos, además de lograr hacer diversas animaciones, tanto sencillas como complejas, utilizando diversas funciones como DoMovement y funciones propias. Además, se realizó el movimiento de cámara para movernos en el espacio.

Logre comprender más sobre los shaders, como lo fueron con las luces y la carga de modelos, además de comprender más a fondo el como se trabaja con OpenGL.

Además de comprender de mejor manera el software de modelado, comprender los mapas UV y las transformaciones básicas dentro del mismo programa.