NATIONAL AUTONOMOUS UNIVERSITY OF MEXICO

ENGINEERING FACULTY

TECHNICAL MANUAL

Group: 04

Teacher:
Roman Balbuena Carlos Aldair

Student:
Mora González Alan Francisco

Semestre 2022-2
19 May 2022

General index

Goal

The student must apply and demonstrate the knowledge acquired throughout the course.

Description

The student must select a facade and a space that can be real or fictitious and present reference images of such spaces for 3D recreation in OpenGL.

In the reference image should be visualized 5 objects that the student will recreate virtually and where such objects should be the most similar to their reference image, as well as their setting.

Scope

It seeks to recreate the lively space: *McDonald's focusing on the exterior facade of the restaurant, as well as the living room along with part of the kitchen, in addition to designing a bathroom for the restaurant in OpenGL using 3d modeling to perform the objects of the environment to be performed with their respective texturing, using Maya Autodesk software to later load such 3D models into OpenGL using hierarchical modeling and geometric transformations. In addition, applying the knowledge of the graphic pipeline such as shaders, camera, drawing, projection, texturing, lighting, etc.*

This space will have interaction with the user through the use of the camera to travel it, as well as simple and complex animations that offer a simple experience of immersion in the context for the same.

Schedule of activities

Activities	Beginning	Duración (días)	Fin
Planning	19 de abril	5	24 de abril
Reference search	19 de abril	4	23 de abril
Creation of the project	24 de abril	1	25 de abril

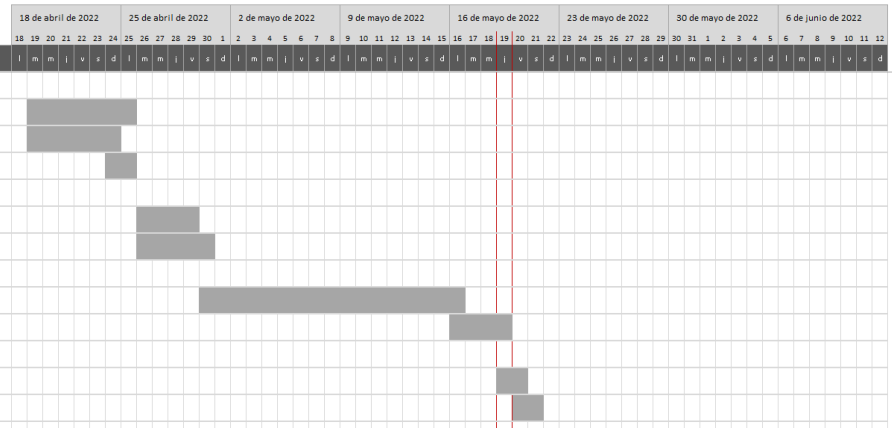
Animations	26 de abril	4	29 de abril
Lighting	26 de abril	5	30 de abril
Modeling and texturing of objects	30 de abril	16	16 May
Object loading	16 May	3	19 May
User manual	19 May	1	20 May
Technical manual	20 May	1	21 May

PROYECTO FINAL

Orange Corp
Alan Francisco Mora

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

TAREA	ASIGNADO A	PROGRESO	INICIO	FIN
Planificación				
Planificacion	Alan Mora	100%	19-4-22	25-4-22
Busqueda de imágenes	Alan Mora	100%	19-4-22	24-4-22
Creacion del proyecto	Alan Mora	100%	24-4-22	25-4-22
Animaciones				
Animaciones	Alan Mora	100%	26-4-22	29-4-22
Iluminación	Alan Mora	100%	26-4-22	30-4-22
Modelado				
Modelado y texturizado de objetos	Alan Mora	100%	30-4-22	16-5-22
Carga de objetos	Alan Mora	100%	16-5-22	19-5-22
Manuales				
Manual de usuario	Alan Mora	100%	19-5-22	20-5-22
Manual tecnico	Alan Mora	100%	20-5-22	21-5-22



Reference images

Objects to be recreated:

- Mesa
- Silla
- Sillón
- Máquina de auto cobro
- Lámparas
- Bote de basura
- Order bar

Objects to be recreated:

- Lavabo
- Soap dispenser
- Tasa de baño
- Hand dryer
- Barra de lavabos

Code

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	Description
GLEW, GLFW	For OS communication and window management in Visual Studio and OpenGL (creation, deletion, etc.).
GLM	C++ mathematical library for OpenGL API projects.
Assimp	Library to import 3d models of programs such as 3ds MAX, Maya, Blender, etc. and that OpenGL can understand their format, in our case allows importing models . obj and read . mtl files with information from these objects to abstract to OpenGL.
SOIL2	It allows to load textures (images) in OpenGL.

Include files

Biblioteca	Description
Shader.h	It creates the vertex shader and fragment shader, the only thing it receives are two addresses, the first for the vertex shader file and the second with the fragment shader file. Open these files by retrieving the code they contain and then create both shader assigning them memory locations, compile them, verify that they have been created correctly.
Camera.h	Create the synthetic or interactive camera, this one. h defines the movements that the camera will have (forward, backward, left and right), the initial positions or values of the camera for rotation, inclination, speed of movement, sensitivity and zoom or angle of opening for the projection in perspective.
Model.h	Get the address of the file. obj and is responsible for importing such a model using assimp, loading coordinates of its vertices (positions, normal and textures).

Shaders

Shaders	Description
lamp.vs y lamp.frag	The vertex accesses the VAO memory location with the position vertex information and affects the arrays with the position, the fragment displays the color information.
lighting.vs y lighting.frag	The vertex shader accesses all the coordinates of the VAO (position, color and texture), receives the matrices of model, view and projection to modify them according to the geometry received from the VAO. The fragment shader defines the types of light that will be available (directional, point and spot) in the form of a structure with its components.

Functions

```
// 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();
```

Shaders	Description
KeyCallback	It processes input by keyboard, keys pressed to perform an action. 5 keys are considered specifically for the interaction with the virtual environment [ESC, W, A, S, D]. Key events 0 to 1023 are also considered.
MouseCallback	It's the mouse input listener. The defining action for this input is the update in the camera rotation.
DoMovement	This function defines the actions to be performed if there are activated animations started in KeyCallback, when they change status and also the update of the camera.
Chairs animations	Within this animation is the necessary programming to make the complex animation of the restaurant chairs.
animationBurger	Within this animation is the programming needed to make the complex animation of the parabolic shot using a burger restaurant.
main	Main function with the program to set the virtual environment. Create the window, check for errors, call the "listener" functions or callbacks for keyboard and mouse events. Defines the viewport, activates OpenGL options for handling translucent and/or transparent materials. Create and configure the shaders. Captures data from motion listeners and keyboard and mouse inputs. It defines the lighting.

Global variables

Variable	Description
camera	Camera type with the initial position of the camera in space.
keys	Arrangement of booleans to handle keyboard input.
firstMouse	Boolean for camera handling.
WIDTH	Variable with window width.
HEIGHT	Variable with window height.
SCREEN_WIDTH y SCREEEN_HEIGHT	To store the width and height that returns some function.
transChair, rotChair and activeChair	Variables for simple chair animation.
transTrash, rotTrash y activeTras	Variables for simple trash can animation.
firstMouse	Boolean for camera handling.
movKitXChair, movKitYChair y rotKitChair	Variables for complex animation of chairs.
circuitoChair (1, 2)	Boolean variables to enable and disable the complex animation of the chairs, depending on these variables and the path that is activated will be the position of the chairs at that time.

vinY, vinZ y rotKitBur	Variables for complex animation of the burger.
circuitoBur (1, 2)	Boolean variables to enable and disable the complex animation of the burger, depending on these variables and the path that is activated will be the position of the burger at that time.
travel (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	Boolean variables for the states of complex animations defined in KeyCallback.
lightPos	Attribute position of lighting.
pointLightPositions	Arrangement with coordinates of 5 point lights.
deltaTime	Time between the current frame and the last frame.
lastFrame	Time of last frame.

Loading of models (Objects).

Models are loaded through the creation of a Model Object for each model. Using Model. h specifying the file path for each model. obj for the library or template to load the models as explained in the include files section.

```
// Models
Model chair((char*)"Models/Silla/silla.obj");
Model table((char*)"Models/Table/table.obj");
Model sofa((char*)"Models/Sofa/sofa.obj");
Model counter(((char*)"Models/Counter/counter.obj");
Model trash((char*)"Models/Trash/trash.obj");
Model tapa((char*)"Models/Trash/tapa.obj");
Model machine((char*)"Models/Machine/machine.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/Restaurant/left.obj");
Model right((char*)"Models/Restaurante/right.obj");
Model ceiling(((char*)"Models/Restaurant/ceiling.obj");

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

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

Animations

The activation of the animations can be found in the KeyCallback function:

```
    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);
```

```
            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;
}
}
```

The first three animations consist of rotating and moving objects, the garbage cans and chairs of the restaurant were animated. In addition, it was made as a simple animation an on and off lights.

The other two animations are complex, the movement of chairs makes transformations a little more complex than the simple ones, by measuring the rotations and translations necessary to simulate the lifting of the chairs. While the jump that makes the burger makes use of the parabolic shot.

Cost analysis

The project includes the salaries of various people who will take over tasks according to their position within the company. These are broken down in the following table (considering that a working day has 7 hours).

Post	Daily wage	Working hours	Working days	Total
Project leader	$1060	42	6	$6360
Líder comercial	$950	21	3	$2850
Diseñador	$800	112	16	$12800
Programmer	$700	91	13	$9100
Total				**$31100**

Human resources

Considering 10% slack and 60% utility

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

$$RH = \$52870$$

Services

- Internet.
  - $700 al mes
- Electrical energy.
  - $150 monthly averages

$$Servicios = 700 + 150$$

$$Servicios = \$850$$

Support tools

- Papelería.
  - $150 a month.

$$Papeleria = \$150$$

Net payment (PN)

$$PN = RH + Servicios + Papeleria$$
$$PN = 52870 + 850 + 150$$
$$PN = \$53870$$

So the total price of the project is four weeks of work equivalent to one month, only having what was agreed in the reference images. You will have to pay a proportional part of 30% to start with it and release the total of the project at the end of the month, along with the delivery. $\$53,870.00$

Conclusions

During this project I could apply the concepts seen in class throughout the course, I understood about modeling and texturing objects, as well as managing to make various animations, both simple and complex, using various functions such as DoMovement and own functions. In addition, the camera movement was performed to move us in space.

Gain a deeper understanding of shaders, such as lighting and model loading, and how OpenGL works.

In addition to better understanding modeling software, understand UV maps and basic transformations within the same program.