

**Proyecto Final  
Manual Técnico**

**Índice**

<b>Objetivo .....</b>	<b>2</b>
<b>Alcance del proyecto .....</b>	<b>2</b>
<b>Cronograma de actividades .....</b>	<b>2</b>
<b>Limitantes .....</b>	<b>3</b>
<b>Documentación del código.....</b>	<b>3</b>
Programa principal – ProyectoFinal.cpp.....	3
Bibliotecas .....	4
Camera.h.....	4
Shaders .....	4
modelLoading .....	4
lamp.....	4
anim .....	4
lighting.....	4
<b>Animaciones.....</b>	<b>4</b>
Animaciones sencillas .....	4
Animaciones complejas .....	5
<b>Repositorio .....</b>	<b>5</b>
<b>Conclusión .....</b>	<b>5</b>

## Objetivo

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

## Alcance del proyecto

Recrear en forma 3D en OpenGL una fachada y un espacio ficticio basada en el anime de Demon Slayer, modelando la “Casa de Urokodaki”, así como 7 objetos similares a los referenciados, además de desarrollar 3 animaciones sencillas y 2 complejas dentro del contexto del espacio generado.

## Cronograma de actividades

El desarrollo del proyecto se llevó a cabo bajo el concepto del “Modelo Incremental” de Software, el cual se caracteriza por reducir tiempos de desarrollo inicial, así como presentar entregas tempranas de partes operativas del producto final.

Actividades	Semana 1					Semana 2					Semana 3					Semana 4				
	28 febrero – 4 marzo					7 – 11 marzo					14 – 18 marzo					21 – 25 marzo				
	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V
Asignación del proyecto																				
Búsqueda y documentación de referencias																				
Aprobación del proyecto																				
Modelado de objetos																				

Actividades	Semana 5					Semana 6					Semana 7					Semana 8				
	28 marzo – 1 abril					4 – 8 abril					11 – 15 abril					18 – 22 abril				
	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V
Modelado de objetos																				
Carga inicial del proyecto a GitHub																				
Modelado de fachada																				

Actividades	Semana 9					Semana 10					Semana 11				
	25 – 29 abril					2 – 6 mayo					9 – 13 mayo				
	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V
Modelado de objetos															
Modelado de fachada															
Implementación de animaciones															
Elaboración de documentación															
Creación del ejecutable															
Entrega del proyecto															

Las entregas mencionadas en la forma del desarrollo se realizaron a partir de la semana 4, dado que cada objeto se presentó una visualización correcta al termino de cada semana.

## Limitantes

La principal limitante del desarrollo del proyecto fue principalmente el tiempo, dado que las horas dedicadas en ocasiones fueron menores a las asignadas en la programación del proyecto, provocando a su vez atrasos en el término de las actividades, así como reducción en la calidad de algunos modelos.

Por otro lado, existe la limitante de las habilidades de diseño y desarrollo, las cuales, aunque fueron mejorando en el proceso, implicaron más tiempo del planeado el adquirir conocimientos y adaptarse al software, principalmente de modelado.

## Documentación del código

La documentación a detalle del programa principal se encuentra dentro del código, en este apartado se mencionará de forma general el contenido y funcionamiento de las partes que componen al proyecto.

### Programa principal – ProyectoFinal.cpp

Contiene toda la estructura del proyecto, primeramente, incluye las bibliotecas necesarias para el correcto funcionamiento del programa además de que se definen constantes y variables. Se realiza la configuración de la ventana, la cual muestra el resultado, se configuran los shaders, VBO, VAO y EBO, posteriormente se cargan los modelos y texturas de los objetos necesarios para modelar el proyecto, se dibujan estos objetos por medio de sus shaders correspondientes, controla la posición de la cámara, y contiene el diseño de las animaciones sencillas.

## Bibliotecas

### Camera.h

Maneja la sensibilidad, movimiento, velocidad, posición, matrices de la cámara.

### stb\_image.h

Maneja la carga de imagenes.

### Shaders

### modelLoading

- .vs: Lectura de la posición de vértices, normales, coordenadas de textura, matrices de modelo, vista y proyección, además envía coordenadas de textura al fragment shader.
- .frag: Recibe coordenadas de textura y manda la textura como color.

### lamp

lamp y lamp2 tienen el mismo funcionamiento, sin embargo, se emplean en diferentes partes del proyecto principal.

- .vs: Contiene las matrices de proyección, vista, modelo y posición.
- .frag: Se envía un color blanco.

### anim

anim, anim2 y anim3 tienen un funcionamiento similar, sin embargo, tienen diferentes funciones y valores, dado que animan objetos separados.

- .vs: Recibe matrices de proyección, vista, modelo y tiempo del procesador, se agregan los movimientos en las posiciones y las coordenadas de textura.
- .frag: Recibe coordenadas de textura y manda la textura como color.

### lighting

- .vs: Lectura de la posición de vértices, normales, coordenadas de textura, trabaja con la matriz de modelo.
- .frag: Recibe componentes ambiental, difusa, especular y el brillo, para las cuales realiza los cálculos necesarios, así mismo trabaja la posición de la luz.

## Animaciones

Detallando las animaciones implementadas.

### Animaciones sencillas

Las animaciones sencillas agregadas al proyecto fueron:

- Apertura y cierre de la puerta principal de la casa.
- Apertura y cierre de la puerta de la habitación.
- Apertura y cierre de las puertas del armario.
- Movimiento en el cajón del armario.

Estas animaciones implican traslaciones en los objetos.

### Animaciones complejas

Las animaciones complejas agregadas al proyecto fueron:

- Movimiento en el fuego de la fogata.
- Movimiento en el caldo dentro de la olla y los ingredientes.

Estas animaciones implican el movimiento de posiciones de los vértices y en las coordenadas de las texturas.

### Repositorio

Se anexa el enlace del repositorio de GitHub en el cual se encuentra el proyecto.

[https://github.com/NellyPLopezSosa/316328667\\_PROYECTO\\_GPO12.git](https://github.com/NellyPLopezSosa/316328667_PROYECTO_GPO12.git)

### Conclusión

Con el desarrollo de este proyecto fue posible aplicar de una forma global los conceptos vistos en las prácticas, obteniendo un resultado bastante completo.

Lo que mejor aprendí de este trabajo fue el uso del software de modelado Maya, el cual resulta bastante completo al contar con diferentes herramientas que enriquecen los objetos, por otro lado, al exportar los modelos en OpenGL fue más sencillo el acomodarlos, aplicarles animaciones sencillas y complejas para que al final tuviera sentido el proyecto.

En contraparte el concepto que fue más difícil de implementar de forma ideal fue la iluminación, por lo cual se descartaron ciertos elementos del resultado final.

**Final project  
Technical manual**

**Index**

<b>Objective .....</b>	<b>2</b>
<b>Project scope .....</b>	<b>2</b>
<b>Schedule of activities.....</b>	<b>2</b>
<b>Limitations.....</b>	<b>3</b>
<b>Code Documentation .....</b>	<b>3</b>
Main Program – FinalProject.cpp .....	3
Libraries.....	3
Camera.h.....	3
shaders.....	4
modelLoading .....	4
lamp.....	4
anim .....	4
lighting.....	4
<b>Animations.....</b>	<b>4</b>
Simple animations.....	4
Complex animations .....	4
<b>Repository.....</b>	<b>5</b>
<b>Conclusion .....</b>	<b>5</b>

## Objective

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

## Project scope

Recreate in a 3D model in OpenGL a facade and a fictitious space based on Demon Slayer anime, modeling "Urokodaki's House", also 7 objects similar to the ones that have been referenced, additionally developing 3 simple and 2 complex animations within the context of the generated space.

### Schedule of activities

The development of the project was carried out under the concept of "Incremental Model" of Software, which is characterized by reducing initial development times, as well as presenting early presentations of operational parts of the final product.

Activities	Week 1					Week 2					Week 3					Week 4				
	February 28 – March 4					March 7 – 11					March 14 – 18					March 21 – 25				
	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V
Project Assignment																				
Search and documentation for references																				
Project approval																				
Object modeling																				

Activities	Week 5					Week 6					Week 7					Week 8				
	March 28 – April 1					April 4 – 8					April 11 – 15					April 18 – 22				
	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V
Object modeling																				
First project upload to GitHub																				
Facade modeling																				

Activities	Week 9					Week 10					Week 11				
	April 25 – 29					May 2 – 6					May 9 – 13				
	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V
object modeling															
Facade modeling															
Implementation of animations															
Documentation															
Creation of executable															
Project deadline															

Presentations that were mentioned in the form of the development were made weekly since week 4, every object was submitted with a correct visualization.

## Limitations

The main limitation in the development of the project was time, hours dedicated were sometimes less than those assigned in project schedule, causing delays to the completion of activities, as well as a reduction in the quality of some models.

Also, existed limitations with the design and development skills, although they were improving in the process, involved more time than planned to acquire knowledge and adapt to the software, mainly the modeling one.

## Code Documentation

The detailed documentation of main program is found within the code, in this section a general summary of content and operation of parts that make up the project is provided.

### Main Program – FinalProject.cpp

It contains the entire structure of the project, firstly includes necessary libraries for the correct operation of the program, in addition has the definition of constants and variables. Configuration of the window which shows the result is carried out, shaders are loaded, VBO, VAO and EBO are configured, models and textures of objects are loaded, these objects are drawn with their shaders, controls the position of the camera, and contains the design of the simple animations.

### Libraries

#### Camera.h

Manages the sensitivity, movement, speed, position, and camera matrix.

#### stb\_image.h

Loades images.



shaders

modelLoading

- .vs: Reads the position of vertex, normal vectors, texture coordinates, model, view and projection matrix, and send texture coordinates to fragment shader .
- . frag: Receives texture coordinates and output texture as a color.

lamp

lamp and lamp2 have the same behavior, however they are used in different parts of the main project.

- .vs: Contains the projection, view, model and position matrix.
- . frag: Sends white color.

anim

anim , anim2 and anim3 have similar behavior, however they have different functions and values since they animate separate objects.

- .vs: Receives projection, view, model matrix and processor time, adds movement to positions and texture coordinates.
- . frag: Receives texture coordinates and output texture as a color.

lighting

- .vs: Reads the position of vertex, normal vectors, texture coordinates, works with the model matrix.
- . frag: Receives ambient, diffuse, specular and brightness components, for which it performs the necessary calculations, and also works on the position of light.

## **Animations**

Detailing the implemented animations.

Simple animations

The simple animations added to the project were:

- Opening and closing of main door.
- Opening and closing of room's door.
- Opening and closing of closet's doors.
- Movement in the closet's drawer.

These animations involve translations in the objects.

Complex animations

The complex animations added to the project were:

- Movement in the fire of the campfire.
- Movement in the broth inside the pot and the ingredients.

These animations involve moving vertex positions and texture coordinates.

## **Repository**

The link of the GitHub repository in which the project is located is attached.

[https://github.com/NellyPLopezSosa/316328667\\_PROJECT\\_GPO12.git](https://github.com/NellyPLopezSosa/316328667_PROJECT_GPO12.git)

## **Conclusion**

With the development of this project it was possible to apply in a global way the concepts seen in the practices, obtaining a fairly complete result.

I learned from this project how to the use of the Maya modeling software, which is quite complete as it has different tools that improve the objects, exporting the models to OpenGL was easier to accommodate them, finally apply simple and complex animations that make sense was not that difficult.

On the other hand, the concept that was the most difficult to implement in an ideal way was lighting, for which certain elements of the final result were discarded.