

**Proyecto Final
Manual Técnico**

Índice

Objetivo	2
Alcance del proyecto	2
Cronograma de actividades	2
Análisis de costos	3
Personal	3
Recursos humanos	3
Servicios.....	3
Depreciación del equipo	3
Material de apoyo	4
Pago neto	4
Limitantes	4
Documentación del código.....	4
Programa principal – ProyectoFinal.cpp.....	4
Bibliotecas	5
Camera.h.....	5
Shaders	5
modelLoading	5
lamp.....	5
anim	5
lighting.....	5
Animaciones.....	5
Animaciones sencillas	6
Animaciones complejas	6
Repositorio	6
Conclusión	6

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 5 objetos por habitación similares a los referenciados, además de desarrollar 4 animaciones dentro del contexto del espacio generado.

Cronograma de actividades

El desarrollo del proyecto se lleva 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				
	18 – 22 abril					25 – 29 abril					2 – 6 mayo					9 – 13 mayo				
	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																				
Análisis de costos																				
Aprobación del proyecto																				
Modelado de objetos																				
Modelado de fachada																				

[illegible]

Análisis de costos

Personal

López Sosa Nelly Paola se asigna como única responsable del proyecto, desempeñando los roles siguientes:

- Líder de proyecto
- Analista comercial
- Diseñador
- Programador

Se toman 8 horas como un día de trabajo, así mismo se considera la experiencia de la persona para las consideraciones del salario.

Puesto	Salario por día	Horas de trabajo	Días de trabajo	Total
Líder de proyecto	\$790	9	1.125	\$888.75
Analista comercial	\$650	5	0.625	\$406.25
Diseñador	\$350	45	5.625	\$1,968.75
Programador	\$480	54	6.75	\$3,240.00
Total				\$6,503.75

Recursos humanos

	Desglose	Total
Holgura de 7%	$0.07 \times 6,503.75$	\$455.26
Utilidad del 50%	$1.5 \times 6,503.75$	\$9,755.63
Total		\$10,210.89

Servicios

Servicio	Costo bimestral	Costo por proyecto (5 semanas)	Total
Energía eléctrica	\$250	\$156.25	\$156.25
Internet	\$780	\$487.50	\$487.50
Total			\$643.75

Depreciación del equipo

Se considera un periodo de 5 años.

Costo del equipo	Costo anual	Costo mensual	Costo por proyecto (5 semanas)	Total
\$32,000	\$6,400	\$533.33	\$666.66	\$666.66

Material de apoyo

Concepto	Costo
Papelería	\$50
Impresión y fotocopias	\$20
Total	\$70

Pago neto

Concepto	Total
Recursos humanos	\$10,210.89
Servicios	\$643.75
Depreciación del equipo	\$666.66
Material de apoyo	\$70
Total	\$11,591.30

El precio total del proyecto es de \$11,591.30. Es necesario un pago por adelantado del 25% del total del proyecto (\$2,318.26) y liquidar a su entrega.

Limitantes

La principal limitante del desarrollo del proyecto fue 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, es importante estar en la rama 316328667_PROYECTOTEORIA_GPOo4.

https://github.com/NellyPLopezSosa/316328667_PROYECTO_GPO12/tree/316328667_PROYECTOTEORIA_GPOo4

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, así mismo fue posible aplicar conocimientos sobre texturizado para lograr que estos tuvieran una apariencia aproximada a las referencias, siendo necesario hacer modificaciones en los mapas UV, haciendo uso de diferentes tipos de proyección.

Por otro lado, al exportar los modelos en OpenGL fue más sencillo el acomodarlos, siendo necesario aplicar modelado jerárquico para dividir en partes a los objetos y de esta forma poder aplicarles animaciones sencillas y complejas para que al final tuviera sentido el proyecto.

Adicionalmente el manejo de shaders facilita la lectura del proyecto puesto que se vuelven partes especializadas a manejar ciertos elementos, tales como la carga de modelos, matrices, coordenadas y componentes, para con ellos realizar los cálculos pertinentes.

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

Objective	2
Project scope	2
Schedule of activities.....	2
Cost analysis.....	3
Staff.....	3
Human Resources	3
Services	3
Equipment depreciation	3
Support material.....	4
Payment.....	4
Limitations.....	4
Code Documentation	4
Main Program – FinalProject.cpp	4
Libraries.....	4
Camera.h.....	4
shaders.....	5
modelLoading	5
lamp.....	5
anim	5
lighting.....	5
Animations.....	5
Simple animations.....	5
Complex animations	6
Repository.....	6
Conclusion	6

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 5 objects like the ones that have been referenced, additionally developing 4 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				
	18 – 22 April					25 – 29 April					May 2 – 6					May 9 – 13				
	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 of references																				
Cost analysis																				
Project approval																				
Object modeling																				
Facade modeling																				

[illegible]

Cost analysis

Staff

López Sosa Nelly Paola is assigned as the person in charge of the project, performing the following roles:

- Project leader
- Commercial analyst
- Designer
- Programmer

8 hours are taken as a working day, likewise the person's experience is considered for salary considerations.

Position	Salary per day	Work hours	Workdays	Total
Project leader	\$790	9	1.125	\$888.75
Commercial analyst	\$650	5	0.625	\$406.25
Designer	\$350	45	5.625	\$1,968.75
Programmer	\$480	54	6.75	\$3,240.00
Total				\$6,503.75

Human Resources

	Breakdown	Total
7% clearance	$0.07 \times 6,503.75$	\$455.26
50% profit	$1.5 \times 6,503.75$	\$9,755.63
Total		\$10,210.89

Services

Service	Bimonthly cost	Cost per project (5 weeks)	Total
Electric power	\$250	\$156.25	\$156.25
Internet	\$780	\$487.50	\$487.50
Total			\$643.75

Equipment depreciation

A period of 5 years is considered.

Equipment cost	Annual cost	Monthly cost	cost per project (5 weeks)	Total
\$32,000	\$6,400	\$533.33	\$666.66	\$666.66

Support material

Concept	Cost
Stationery material	\$50
Printing and photocopies	\$20
Total	\$70

Payment

Concept	Total
Human Resources	\$10,210.89
Services	\$643.75
Equipment depreciation	\$666.66
Support material	\$70
Total	\$11,591.30

The total price of the project is \$11,591.30. It is necessary an advance payment of 25% of the total project (\$2,318.26) and pay upon delivery.

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, it is important to be in the 316328667_PROYECTOTEORIA_GPO04 branch.

https://github.com/NellyPLopezSosa/316328667_PROYECTO_GPO12/tree/316328667_PROYECTOTEORIA_GPO04

Conclusion

With the development of this project, it was possible to apply in a global way the concepts seen in the practices, obtaining a 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, likewise it was possible to apply knowledge about texturing to make them have an approximate appearance to the references, being necessary to make modifications to the UV maps, making use of different types of projection.

On the other hand, when exporting the models in OpenGL it was easier to accommodate them, being necessary to apply hierarchical modeling to separate the objects into parts and thus be able to apply simple and complex animations so that the project would make sense in the end.

Additionally, the handling of shaders facilitates the reading of the project since they become specialized parts to handle certain elements, such as the loading of models, matrices, coordinates, and components, to carry out the pertinent calculations with them.

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 result were discarded.