

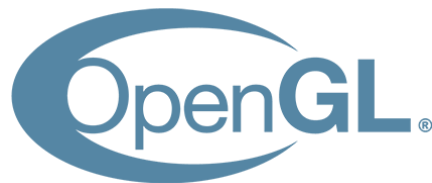


Universidad Nacional Autónoma de México

Facultad de Ingeniería

Proyecto Final

Laboratorio de computación grafica



Nombre: Herrera Godina Diana Celeste

Número de cuenta: 316161927

Grupo: 12

Profesor: Ing.Carlos Aldair Roman Balbuena

Índice

Manual Técnico:

Objetivos	3
Fachada y cuarto a recrear	3
Objetos para recrear	4
Diagrama de Gant	6
Alcance del proyecto	6
Documentación del código	7
Librerías usadas	7
GL includes	8
Shaders	8
Funciones	9
Carga de modelos	9
Animaciones sencillas (variables)	10
Animaciones complejas (variables)	11
Conclusión	15
Enlace github proyecto	15

Proyecto Final

Objetivo

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

Fachada y cuarto a recrear:



Fachada y cuarto en 3D (Software OpenGL):

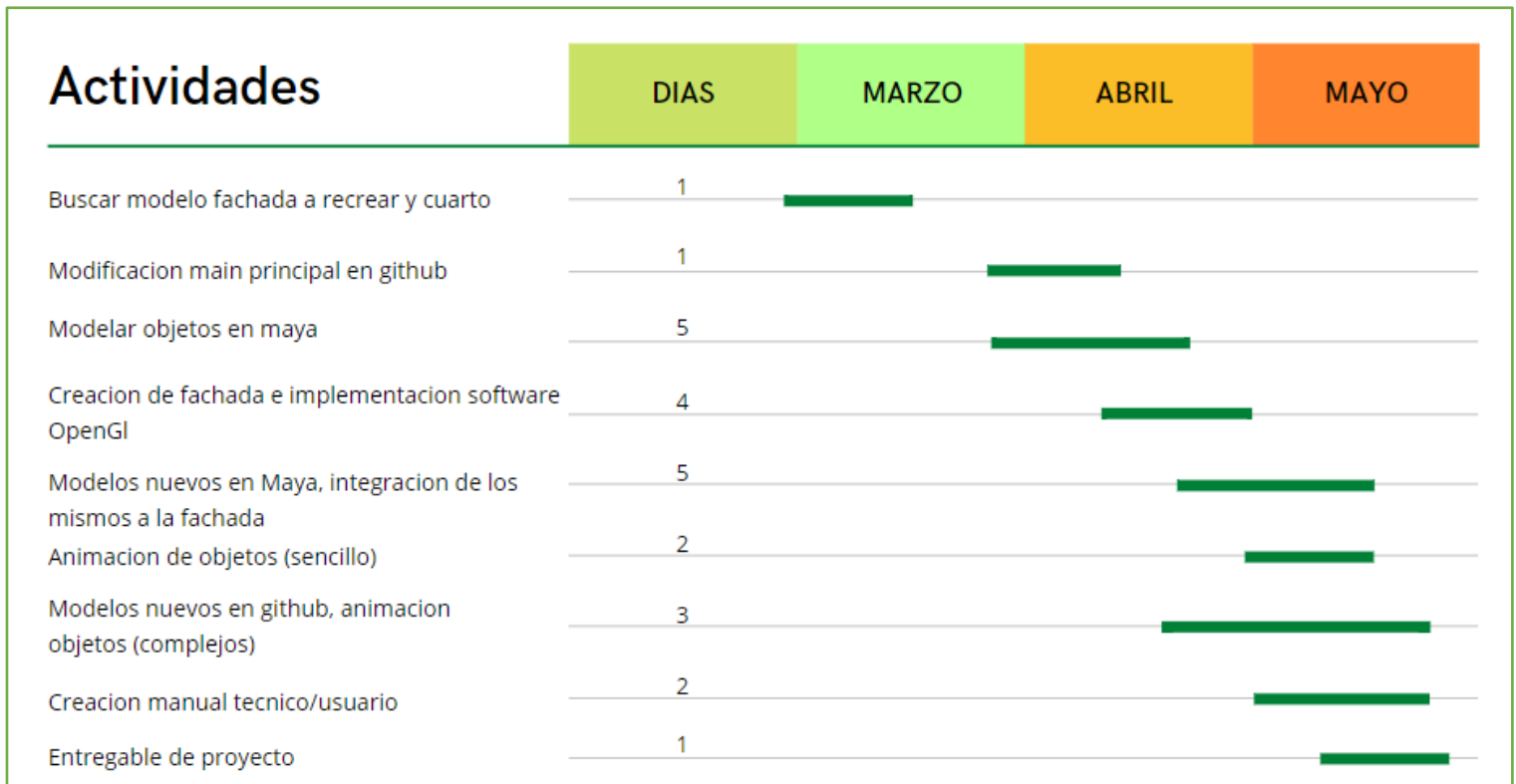


Objetos para recrear dentro de la habitación:

Objetos para recrear	Imagen de referencia	Objeto en OpenGL 3D
Repisas		
Cama		
Escritorio		
Silla		

Jarrón		
Laptop		
Planta		

Diagrama de Gantt



Alcance del proyecto

En el proyecto se busca modelar y recrear objetos lo mas realistas posibles (casa moderna) basándonos en una fachada buscada en internet, una habitación moderna y de esta forma aplicar nuestras habilidades adquiridas en el curso de laboratorio de computación gráfica, adecuando los elementos y llevándolo a la plataforma de versiones GitHub, crear animaciones dentro de este entorno, las actividades que se realizaron llevaron un orden conforme se iba modelando, se requería subir a GitHub con la finalidad de cargar los modelos y tener las versiones dentro de la plataforma.

Implementación: Este proyecto se puede implementar para desarrollo de alguna aplicación o simplemente si se requiere modelar algún espacio que posteriormente se pueda construir o implementar algo tecnológico y poder visualizar de una mejor manera.

Entregables: Se requiere al final tener un archivo ejecutable para entregar este proyecto, además de los manuales de usuario y técnico.

Limitantes: los recursos que se contaron para realizarlo eran pocos por el tipo de computadora esto al final complico un poco la ejecución del programa (OpenGL) ya cargando todos los modelos dentro del main principal, para realizar este proyecto se requirió de algunos meses para ir modificando lo requerido. Los resultados esperados buscan cumplir el objetivo, mostrar el conocimiento adquirido a lo largo del semestre.

Requisitos: Para realizarlo se requiere tener un equipo de computo capaz de manejar el software, OpenGL, Maya, Gimp.

Otros requisitos: Tener conocimiento básico de plataforma GitHub, así crear un repositorio para tener un control de versiones y así en la revisión de pueda descargar el proyecto y posteriormente aprueben nuestro proyecto final.

Documentación del código

Librerías usadas:

Biblioteca	Descripción
GLEW	biblioteca de carga de extensiones C/C++ determina que extensiones de OpenGL son compatibles con la plataforma.
GLFW	Proporciona una API simple para crear ventanas, contextos y superficies, recibir entradas y eventos.
GLM	Implementación matemática de C++.
SOIL2	Carga texturas para los modelos.
Assimp	Importa modelos 3D, en nuestro caso Maya, modelos obj para posteriormente leer la información y moverla a OpenGL.

GL includes:

Includes	Descripción
Shader.h	Sirve para recuperar el código que contiene el vertex y fragment, asignándole localidades de memoria.
Camera.h	Camara sintética sirve para movernos dentro del entorno.
Model.h	Importa los modelos usados “obj” cargando las coordenadas de los vértices de dichos objetos para importarlos, carga las texturas.
stb_image.h	Carga imágenes como texturas, cargando los archivos para integrarlos al proyecto.

Shaders

Shaders	Descripción
anim.frag y anim.vs	Nos ayuda y brinda información para animar dentro del shader.
lighting.vs y lighting.frag	Tiene el control de las coordenadas tanto como de la textura, modifica las matrices de modelo. El frag nos ayuda en la luz con base a sus componentes (espectacular, difusa, todo sobre los materiales).
lamp.vs y lamp.frag	Frag accede a los colores, el vertex revisa las localidades de memoria “VAO” con la información de los vértices tanto como las matrices, revisa dicha información.

Funciones:

Función	Descripción
DoMovement	Si hay animaciones dentro de KellCallback cambian de estado revisando si la animación esta activa.
KeyCallback	Capta la información del teclado, procesa esta información para los eventos.
animacionR	Función para la animación del ratón (juguete) con tiro parabólico. Además de la animación de la regadera (usando seno del tiempo).
MouseCallback	Capta la información del mouse para así rotar y moverlos en el entorno.

Carga de modelos:

```
// Carga de modelos
Model fachada((char*)"Models/Fachada/fachada2.obj");
Model desk((char*)"Models/Desk/desk.obj");
Model cuadro((char*)"Models/Cuadros/cuadros.obj");
Model cama((char*)"Models/Bed/bed.obj");
Model silla((char*)"Models/Silla/silla.obj");
Model aire((char*)"Models/Aire/aire.obj");
Model repisa((char*)"Models/Repisa/repisa.obj");
Model planta((char*)"Models/Planta/planta.obj");
Model lap((char*)"Models/Laptop/laptop.obj");
Model patin((char*)"Models/Patineta/patineta.obj");
Model regadera((char*)"Models/Regadera/regadera.obj");
Model libro((char*)"Models/Libro/book.obj");
Model raton((char*)"Models/Juguete/juguete.obj");
```

Se especifico la ruta para cada modelo especifico dentro de la carpeta "Models" indicando el nombre del obj para que así del archivo model.h pueda cargarlos.

Animaciones sencillas:

Silla de escritorio animación

Variable animación	Tipo	Función
acsilla	bool	Revisa si la función esta activa
transilla	float	Función de translate, la activa
rotsilla	float	Función para rotar, la activa

Patineta animación

Variable animación	Tipo	Función
acpat	bool	Revisa si la función esta activa
transpat	float	Función de translate, la activa
rotpat	float	Función para rotar, la activa

Libro animación

Variable animación	Tipo	Función
transbookX	float	Función de translate para eje X
transbookY	float	Función de translate para eje Y
transbookZ	float	Función de translate para eje Z
rotbook	float	Función para rotar, la activa
acbook	bool	Revisa si la función esta activa

En estas animaciones sencillas solo fueron linealmente, usando funciones: rotación, traslación y escalamiento donde estas se requieren parámetros de la matriz y su vector posición (x,y,z).

Animaciones complejas:

Animación de la regadera:

```
void animacionR()
{
    // Movimiento de la regadera usando la variable tiempo para oscilar
    if (circuitoReg1)
    {
        if (recorrido1)
        {
            movKitYReg += 0.5f;
            if (movKitYReg < 1.5)
            {
                recorrido1 = false;
                recorrido2 = true;
            }
        }
        if (recorrido2)
        {
            rotKitReg = -60;
            movKitZReg -= 0.5f;
            if (movKitZReg < -2.0)
            {
                recorrido2 = false;
                recorrido3 = true;
            }
        }
        if (recorrido3)
        {
            movKitZReg = sin(tiempo + 5);
            movKitXReg = sin(tiempo);
        }
    }
}
```

Regadera

Variable animación	Tipo	Función
movkitXReg	float	Mueve en el eje X
movkitYReg	float	Mueve en el eje Y , se activa para que no llegue hasta el suelo la regadera
movkitZReg	float	Mueve en el eje Z
rotKitReg	float	Función para rotar, la activa para así tenga un grado de inclinación y parezca que la regadera está en un ángulo que alcance las plantas.

tiempo	float	Variable para el tiempo, para aplicar la función de seno y se pueda mover ondulatoriamente.
circuitoReg1	Bool	Función para recorrido
Recorrido1	Bool	Variable para recorrido 1
Recorrido2	Bool	Variable para recorrido 2
Recorrido3	Bool	Variable para recorrido 3; se aplica la función de seno del tiempo + la rotación requerida

Animación del ratón de juguete (carro):

```
//movimiento de ratoncito usando tiro parabolico
if (circuitoRat1)
{
    if (recorrido4)
    {
        rotKitRat = 0;
        v0y += 0.5f;
        v0z += 0.5f;
        if (v0z > 1.5 || v0y > 1.5) //calcule la altura maxima del raton
        {
            recorrido4 = false;
            recorrido5 = true;
        }
    }
    if (recorrido5)
    {
        v0y -= 0.5f;
        v0z += 0.5f;
        if (v0z > 3.5 || v0y < 0.0) //altura va auementando cuando cae
        {
            recorrido5 = false;
            recorrido6 = true;
        }
    }
}

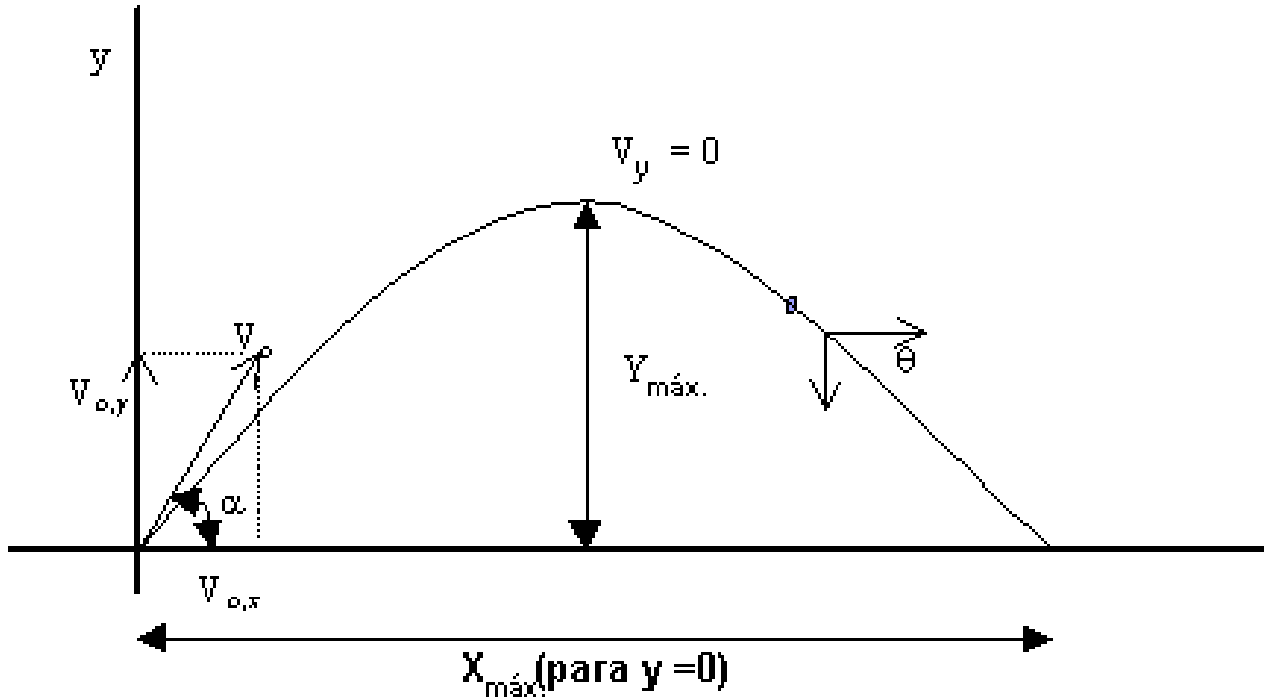
if (circuitoRat2)
{
    if (recorrido6)
    {
        rotKitRat = 180; //regresando el raton a su posicion inicial
        v0z -= 0.5f;
        if (v0z < 0.5)
        {
            recorrido6 = false;
            recorrido4 = true;
        }
    }
}
```

```

    }
}

```

Para la animación del tiro parabólico (ratón)



Utilice para esta función una y_{max} para que así el ratón no se fuera hasta el suelo o hasta el techo, se requirió V_{0y} y una V_{0x} para así podernos mover, ya que al caer este va a cambiar su velocidad, se requiere tener mas fuerza para regresar al suelo y tener la posición original, en este caso el ratón al momento de caer regresara con un recorrido por el suelo “imitando un coche de juguete”.

Ratón de juguete:

Variable animación	Tipo	Función
v_{0y}	float	Mueve en el eje Y en el tiro parabólico, es la altura máxima en este caso.

v0z	float	Mueve en el eje Z, en el tiro parabólico
rotKitRat	float	Rota para que regrese a su posición inicial el ratón.
circuitoRat1	bool	En este recorrido se incrementa tanto x como y para lograr que avance de forma ascendente, realizando una curva y llegando a Ymax.
circuitoRat2	bool	Función para recorrido de regreso para el ratón.
Recorrido4	Bool	Identifica y rota el ratón la altura máxima para incrementar.
Recorrido5	Bool	Decrementa la velocidad en que cae el ratón de juguete para que llegue de el otro lado al bajar la curva pasando por la altura máxima que se puede mover.
Recorrido6	Bool	Mueve el ratón 180° para regresar a un punto y este pueda volver a su posición original.

Conclusiones proyecto:

Los objetivos del proyecto se cumplieron ya que podemos implementar todo lo que vimos a lo largo del curso en OpenGL y en algunas otras herramientas de modelado además de un

editor de imágenes, Este proyecto de igual forma nos sirve para poder implementarlo en la vida real ya que podemos modelar cualquier cosa o proyecto de ingeniería, además pudimos realizar modelos y con base a ellos animar, estas animaciones requirieron conocimientos previos vistos en el curso. Además, pudimos texturizar los modelos y cargarlos a Open gl para así obtener los resultados de la fachada y del cuarto. Recordando todos los conocimientos que aplicamos en el semestre.

Enlace github:

https://github.com/DianaCelesteHerrera/316161927_Proyecto_Gpo12.git