



Universidad Nacional Autónoma de
México



Facultad de Ingeniería

Computación Gráfica e Interacción
Humano Computadora

Semestre 2021-2

Profesor: Carlos Aldair Roman Balbuena

Manual Técnico: Thousand Sunny

Fecha de entrega: 10 de agosto de 2021

Lázaro Martínez Abraham Josué

Índice

Análisis	2
Objetivo	2
Alcance	2
Requerimientos	2
Cubierta del barco y Barco	3
Cocina	4
Objetos a modelar	4
Cronograma del proyecto	5
Herramientas de Hardware	7
Herramientas de Software	7
Análisis económico	7
Diseño	8
Modelado	8
Modelos	9
Personajes	12
Texturas	12
Bibliotecas	16
Interacción	16
Desarrollo	18
Directorios	18
Archivos de cabeceras adicionales	18
Shaders	19
Código principal	20
Diccionario de datos	20
Diccionario de Funciones	22
Experimentos	24
Link Video demostrativo	26
Link proyecto en github	26

Análisis

Objetivo

El presente proyecto tiene como propósito aplicar todos los conceptos vistos durante el curso de computación gráfica e interacción humano-computadora, de forma que al seguir los pasos establecidos principalmente por el pipeline gráfico y el pipeline de renderizado se tenga como resultado final una escena gráfica en la cual se pueda realizar un recorrido virtual.

Alcance

Desarrollo de un software gráfico que permita el recorrido virtual a través de dos cuartos. La escena gráfica contendrá elementos con diferentes tipos de materiales, texturas, animaciones, interacciones y por lo menos un modelo de iluminación.

Requerimientos

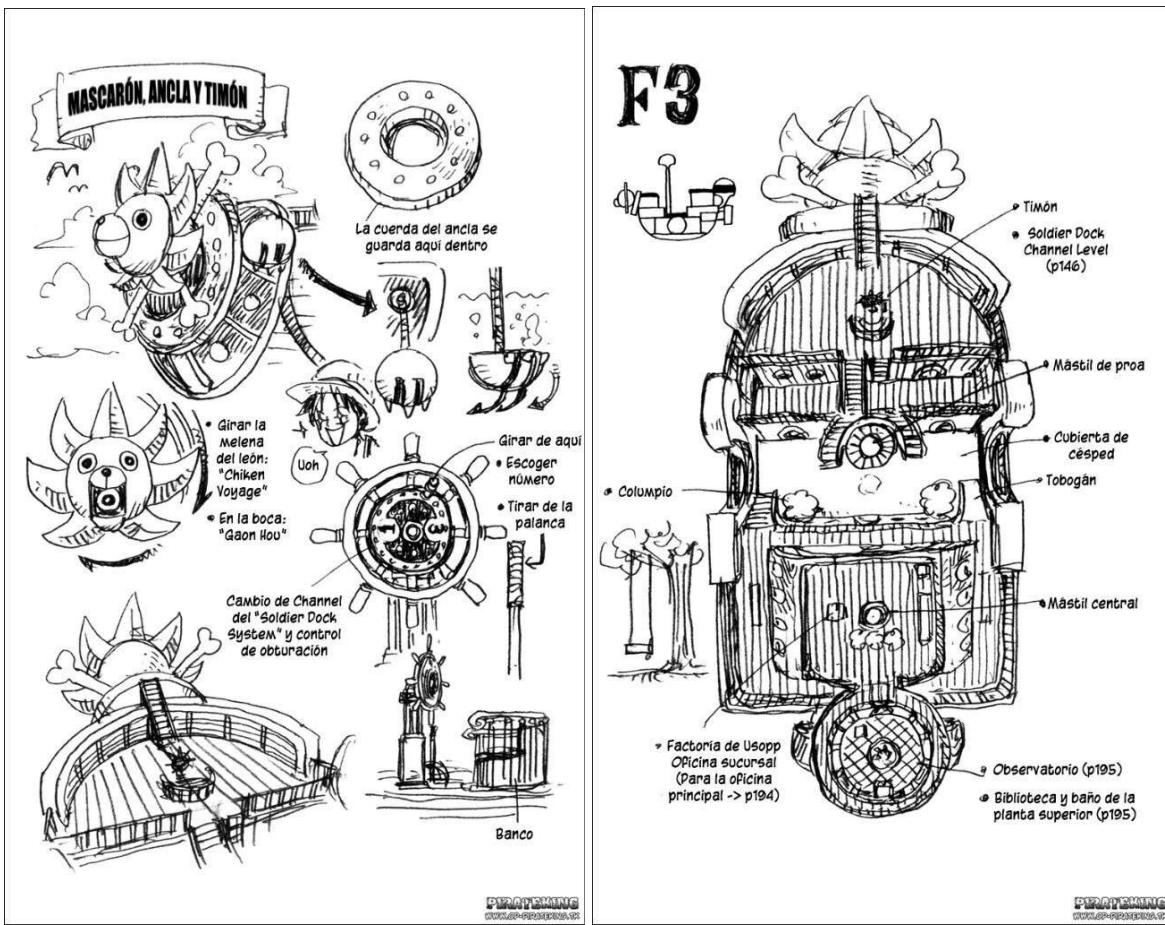
El presente proyecto debe contar con los siguientes requerimientos mínimos:

- ❖ Integrar cámara sintética
- ❖ Contener 4 animaciones diferentes
- ❖ Modelar dos cuartos y una fachada.

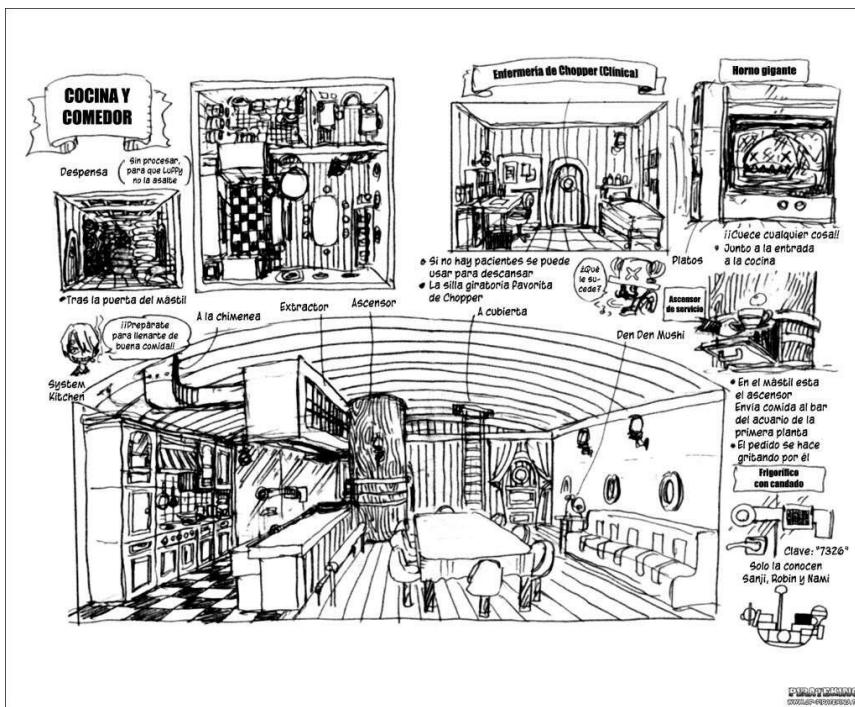
De manera específica, el proyecto recreará el barco Thousand Sunny de la serie animada o anime One Piece con la cubierta con mástiles como primer cuarto y la cocina del barco como segundo cuarto. Contendrá cuatro animaciones: dos animaciones diferentes para dos personajes de la misma serie. Se implementará un skybox para el fondo y dos diferentes modelos de iluminación (Phong y Fresnel).

Imágenes de referencia

Cubierta del barco y Barco



Cocina



Objetos a modelar

Thousand sunny (Barco)

- ❖ Cabeza de león
- ❖ Base de la cabeza de león
- ❖ Mástil principal
- ❖ Mástil secundario
- ❖ Propulsor
- ❖ Escaleras
- ❖ Puertas
- ❖ Ventanas

- ❖ Cuarto externo trasero
- ❖ Anclas en forma de patas
- ❖ Timón
- ❖ Árboles y plantas
- ❖ Curvas de los costados
- ❖ Casco del barco (Base del barco)
- ❖ Cañones laterales
- ❖ Compartimiento con número 1

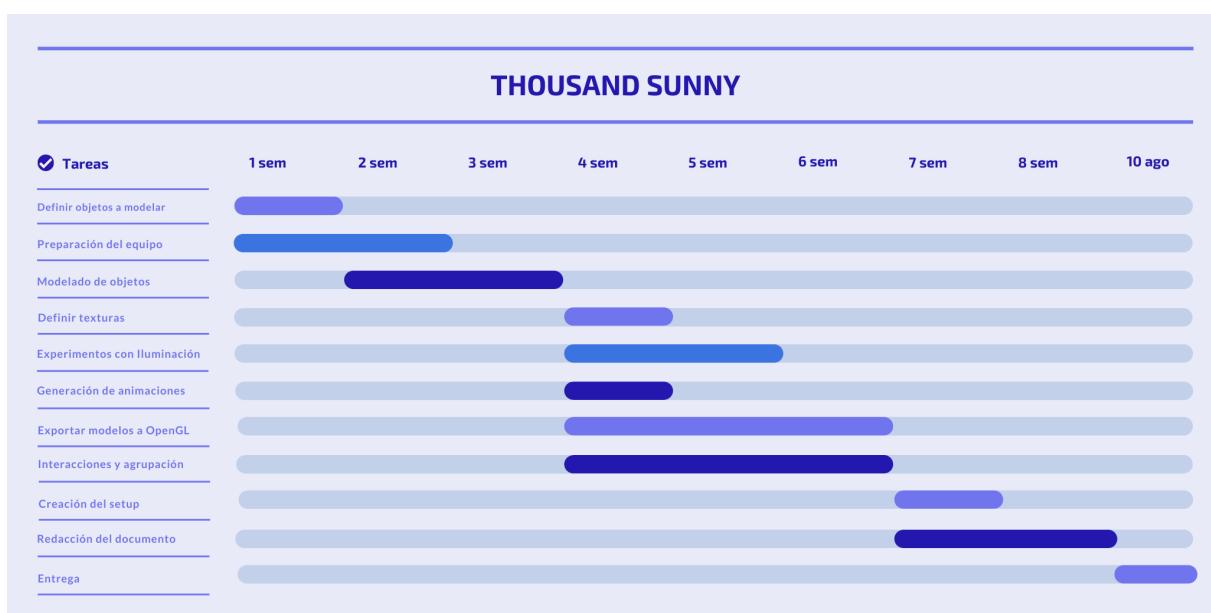
Cocina

- ❖ Mesa
- ❖ Sillas
- ❖ Sofá
- ❖ Extractor
- ❖ Chimenea
- ❖ Barra de comida
- ❖ Estufa
- ❖ Estantes
- ❖ Sartenes
- ❖ Refrigerador
- ❖ Tronco pilar
- ❖ Mesilla para el sofá

Cronograma del proyecto

Para cualquier proyecto, un análisis del tiempo que tomará el desarrollo de este siempre es necesario. Más aún, tomando en cuenta el recorte de tiempo generado a raíz de los problemas suscitados en la facultad, el tiempo es el punto clave para el desarrollo del proyecto.

A continuación, en el diagrama, podemos observar el diagrama de Gantt propuesto para la elaboración de este proyecto.



❖ Definir objetos a modelar

En esta etapa se definirá el alcance del proyecto, estableciendo los entregado en el documento de referencias. Esto nos servirá como una guía para determinar qué modelos se realizarán.

❖ Preparación del equipo

En esta etapa el equipo se preparará, tomando cursos rápidos sobre el manejo del software de modelado, en este caso Blender, y sobre la utilización de los shaders y OpenGL en C++.

❖ Modelado de objetos

Se tomará dos semanas completas para el desarrollo de los modelos en blender. El equipo dividirá los objetos a modelar para la rápida elaboración de estos. Durante esta etapa se ahorrará mucho tiempo gracias a la etapa anterior.

❖ Definir texturas

Ya con los modelos obtenidos, se definirán las texturas para los materiales que los llevarán, y los materiales (componentes difusos, especulares y ambientales) para los que llevarán solo colores.

❖ Experimentos con iluminación

Se dividirán los materiales dependiendo de sus características como los plásticos, los metales, los materiales mates, los materiales translúcidos, para modelos dinámicos (animaciones) y estáticos. También se desarrollarán experimentos para obtener la iluminación adecuada combinando diferentes tipos de iluminación.

❖ Generación de animaciones

Se definirán los modelos de los personajes que se utilizaran, ya sean modelados u obtenidos de internet, y sus animaciones, utilizando la plataforma de Mixamo.

❖ Exportar modelos a OpenGL

Se exportarán los modelos a OpenGL con los shaders adecuados definidos en la parte de experimentación. Los materiales serán exportados desde Blender para asegurar la correcta exportación de las animaciones y de los modelos estáticos con los mismos tipos de materiales.

❖ Interacciones y agrupación

Se terminará de desarrollar el proyecto creando las interacciones con el modelo por medio del ratón y el teclado. Se agruparán las operaciones similares para poder realizar un código mucho más limpio y estructurado. Se realizarán las correcciones necesarias al proyecto para su operación sin errores.

❖ Creación del setup

Una vez compilado el proyecto, este se empaquetara utilizando install Forge. De esta manera el proyecto tendrá una presentación más formal. El tiempo destinado a esta etapa también involucra el estudio y aprendizaje de la herramienta.

❖ Redacción del documento

Se realizará el documento de documentación y manual técnico, así como el documento que servirá como manual de usuario.

❖ Entrega

Entrega del proyecto

Herramientas de Hardware

- ❖ **Equipo:** Laptop
- ❖ **Procesador:** Intel Core i7-8750H 2.20 GHz
- ❖ **Memoria Ram:** 16 GB DDR4-2666 SDRAM
- ❖ **Tarjeta gráfica:** NVIDIA GeForce GTX 1050 ti

Herramientas de Software

- ❖ Blender para realizar nuestros modelos, animarlos, aplicar texturas y previsualización de colores.
- ❖ Mixamo para animar los modelos que representan a los personajes del proyecto.
- ❖ Sketchfab para buscar modelos gratis que representan personajes del entorno desarrollado.
- ❖ API OpenGL en la versión 3.3, para los lenguajes C/C++, así como GLSL para realizar el código como los shaders.
- ❖ Microsoft Visual Studio 2019 y Sublime Text para la edición de código.
- ❖ Utilizamos bibliotecas como GLFW, GLM, GLAD y también ASSIMP. ASSIMP es una biblioteca que nos permite cargar archivos de modelado almacenados en gran variedad de formatos, como: Collada (*.dae; *.xml), Blender (*.blend), 3D Studio Max 3DS (*.3ds), Wavefront Object (*.obj), entre otros. ASSIMP puede cargar información de vértices, coordenadas de textura, normales, materiales, animación, y otros.

Análisis económico

Tomando en cuenta el diagrama de Gantt con los tiempos esperados del proyecto, podemos observar que se contemplan 3 semanas laborales para diseñadores 3D y 5 semanas laborales para desarrolladores de software. A su vez, hay que contemplar el tiempo para el líder de proyecto, el cual debe estar presente desde la semana 1 para definir los alcances del proyecto, hasta la última semana, donde se desarrollará la documentación y entrega del proyecto.

Contemplando una jornada laboral normal, 5 días a la semana, los diseñadores trabajarán 15 días, los desarrolladores 25 y el líder de proyecto 40. Se contratarán dos desarrolladores y dos diseñadores, junto con el líder de proyecto.

Respecto a los materiales de trabajo, al utilizar Blender, que es un software de libre uso, no hay necesidad de pagar membresías para utilizarlo. La misma situación se encuentra con el uso de la plataforma Mixamo, Sketchfab y Visual Studio 2019. Se incluye el costo relativo a 5 equipos de cómputo, ya que el equipo de trabajo ya cuenta con equipo, por lo cual no es necesario comprar equipo nuevo.

Es necesario considerar costos fijos como electricidad e internet, los cuales se contemplarán para las 8 semanas como 2 meses.

Rubro	Especificación	Costo	Total
Sueldo disenador 3D (2)	Pago por 15 días	\$500.00	\$15,000.00
Sueldo desarrollador (2)	Pago por 25 días	\$600.00	\$30,000.00
Sueldo lider proyecto	Pago 40 días	\$600.00	\$24,000.00
Equipo de cómputo	Equipos mencionados	\$1,000.00	\$5,000.00
Licencias de Software	Blender, Sketchfab, Mixamo, VB 2019	\$0.00	\$0.00
Costos Fijos	Electricidad, 2 meses Internet, 2 meses	\$200.00 \$350.00	\$400.00 \$700.00
Total			\$70,100.00

Es importante tomar en cuenta que los tiempos aquí establecidos son considerando desarrolladores con poca experiencia, y que desarrolladores y diseñadores profesionales necesitan menos tiempo para realizar las tareas de cada uno, por lo cual los costos podrían variar.

Diseño

Modelado

El modelo del barco y la cocina fueron realizados en blender utilizando una técnica conocida como Box Modeling, la cual consiste en tener una

figura base (la mayoría de las veces, un cubo) y extruir las caras de esta figura para darle las formas deseadas. Los modelos de los personajes fueron exportados de la página sketchfab con modelos de acceso público.

Cada archivo del modelo se encuentra en la carpeta `bin/models/Sunny`. El modelo se juntó y se dividió por materiales para poder modificar cada uno de los materiales y que estos se vean consistentes entre todos los objetos que tengan el mismo material.

Dentro del directorio encontramos 76 archivos en formato FBX para cada material distinto. De dichos modelos, 15 cuentan con texturas, las cuales podemos encontrar en una carpeta con el mismo nombre del modelo con terminación `.fbm`.

Modelos

Archivo	Descripción
AmarilloTecho.fbx	Techo color amarillo del cuarto trasero y cuarto del mastil
BandaSuperior.fbx	Banda roja del primer piso, parte exterior
BanderaRoja.fbx	Mastil de color rojo
BarandalesYLaterales.fbx	Barandales, escaleras y soporte del cuarto trasero
BarcoBase.fbx	Color de los mástiles
BarcoBaseTextura.fbx	Base del barco (casco), cuenta con textura
BarcoCostado.fbx	Costado circular con número 1, cuenta con textura
BarraCuerpo.fbx	Cuerpo de la primera barra de la cocina
BarraMesa.fbx	Superficie de la primera barra de la cocina
BarraPiso.fbx	Parte inferior de la primera barra de la cocina
BaseBandera.fbx	Base bandera del mástil trasero
BaseCuartoMastil.fbx	Base del cuarto del mástil delantero
BaseSilla.fbx	Base de las sillas de la cocina
BotonesDorados.fbx	Botones y anclas doradas del barco
Campana1.fbx	Color blanco de la campana de la cocina
Campana2.fbx	Color azul de la campana de la cocina
CampanaFondo.fbx	Parte interior de la campana
CampanaTope.fbx	Parte superior de la campana

Canons.fbx	Marco blanco de los costados y cañones laterales
CintalInferior.fbx	Banda negra de la planta baja exterior
CostadosRojos.fbx	Plástico rojo de los costados
CubeMap.fbx	Fondo, cuenta con textura
EstufaCuerpo.fbx	Cuerpo de la estufa secundaria de la cocina
EstufaIntegral.fbx	Cuerpo de la estufa integral y estantes
EstufaIntegralMarco.fbx	Interior de la estufa integral y estantes
EstufaMesa.fbx	Superficie de la estufa secundaria de la cocina
EstufaPiso.fbx	Parte inferior de la estufa secundaria de la cocina
Extractor.fbx	Color cuerpo extractor
ExtractorFondo.fbx	Interior del extractor
ExtractorMarco.fbx	Marco del interior del extractor
FondoCanons.fbx	Centro e interior de los cañones
FondoCara.fbx	Cabeza del león y parte amarilla de la base del león
FondoOjos.fbx	Color de los ojos del león
Hojas.fbx	Color de las hojas de los árboles
Huesos.fbx	Huesos detrás de la cabeza del león
MaderaArbol.fbx	Madera de los árboles
MaderaPuerta.fbx	Madera de las puertas
MangoEstufa.fbx	Mango de las estufas de la estufa integral
Mantel.fbx	Mantel de la mesa de la cocina, cuenta con textura
MarcoCostado.fbx	Marco de la sección lateral del costado con el número 1
MarcoFarol.fbx	Marco y base de los faros traseros
MarcoOroPuerta.fbx	Marco dorado de algunas puertas y ventanas
MarcoPuertaBlanco.fbx	Parte blanca de las puertas con marco
MarcoPuertaRojo.fbx	Parte roja de las puertas con marco
MarcoVentana.fbx	Marco de las ventanas del cuarto trasero

MaterialCadena.fbx	Cadenas del columpio del árbol principal
Melena.fbx	Melena del león
Mesa.fbx	Madera de la base y patas de la mesa, cuenta con textura
Mesita.fbx	Madera de la mesa que está a lado del sofá
MetalMastil.fbx	Metal de los mastiles
Paredes1.fbx	Paredes de los cuartos con color crema
ParedesCocina.fbx	Paredes azules de la cocina
ParedMadera.fbx	Pared trasera de madera de la cocina, cuenta con textura
Pasto.fbx	Pasto de la cubierta del barco, cuenta con textura
PisoAjedrez.fbx	Piso de la cocina de la cocina, cuenta con textura
PisoMadera.fbx	Piso de madera del barco, exterior, cuenta con textura
PisoMadera1.fbx	Piso de madera de la cocina, cuenta con textura
Propulsor.fbx	Propulsor de la parte trasera del barco
Refri.fbx	Refrigerador de la cocina
RefriMarco.fbx	Marcos oscuros del refrigerador de la cocina
RejillaVentana.fbx	Marco de las ventanas cuadradas del exterior
Sarten.fbx	Sartenes de la cocina
Silla.fbx	Color principal de las sillas de la cocina
SillaInterior.fbx	Interior de las sillas de la cocina
Soffa1.fbx	Cojines del sofá, cuenta con textura
Soffa2.fbx	Base de madera del sofá, cuenta con textura
TechoCocina.fbx	Techo rojo de la cocina
TechoParedSuperior.fbx	Paredes/techo rojo del primer piso, exterior de la cocina
Timon.fbx	Timón del barco
TransicionTechoCocina.fbx	Parte blanca entre la pared azul y el techo rojo de la cocina
Tronco.fbx	Tronco pilar de la cocina, cuenta con textura

TroncoSaliente.fbx	Parte saliente del tronco pilar de la cocina
Velas.fbx	Velas blancas y banderas del barco
VentanaEstufa.fbx	Ventana de vidrio de la cocina integral
VentanaPuertaSinMarco.fbx	Ventanas de las puertas sin marcos
Vidrio.fbx	Vidrio de las demás ventanas y faros

Personajes

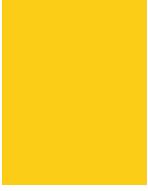
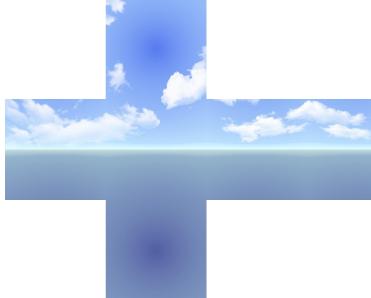
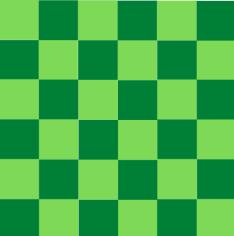
Los modelos de los personajes se encuentran en la carpeta `bin/models/Sunny/Animaciones`. Los personajes (Zoro y Robin) son personajes de la serie One Piece. Los modelos fueron descargados de manera gratuita en la página Sketchfab y animados en Mixamo.

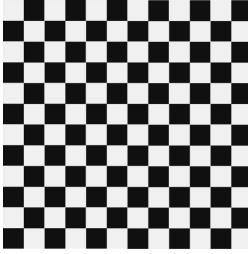
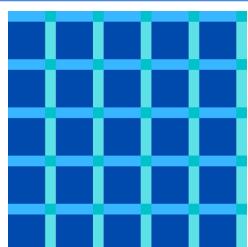
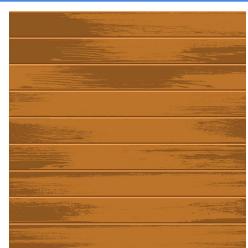
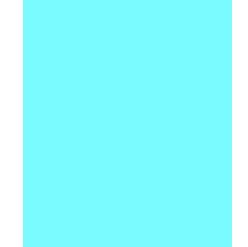
Archivo	Descripción
Zoro.fbx	Archivo con animaciones, mallas y huesos del personaje Zoro, cuenta con texturas
Robin.fbx	Archivo con animaciones, mallas y huesos del personaje Robin, cuenta con texturas

Texturas

Las texturas de los modelos se encuentran al mismo nivel de carpetas que su respectivo archivo FBX.

Carpeta	Imagen	Nombre imagen
BarcoBaseTextura.fbm		MaderaBaseBarco.jpg
BarcoCostado.fbm		uno.jpg

BotonesDorados.fbm		oro.jpg
CubeMap.fbm		Skybox.jpeg
MaderaArbol.fbm		arbol.png
Mantel.fbm		Mantel.png
ParedMadera.fbm		arbol.png
Pasto.fbm		cesped.jpg

PisoAjedrez.fbm		ajedrez2.png
PisoMadera.fbm		MaderaPiso2.jpg
PisoMadera1.fbm		MaderaPiso2.jpg
Soffa1.fb		sofa.png
Soffa2.fbm		MaderaBaseBarco.jpg
Tronco.fbm		arbol.png
Vidrio.fbm		7afbfff.png

Robin.fbm			eye_d.jpg y robin_d.jpg
Zoro.fbm	A close-up view of Zoro's face, showing his eyes and mouth, wearing a green headband and a red bandana.	A close-up view of Zoro's face, showing his eyes and mouth, wearing a green headband and a red bandana.	r.jpg

También se utilizaron las caras del cubemap separado para implementar la iluminación de Fresnel, estas caras se encuentran en la carpeta `bin/models/CubeMapFondo`.

Archivo	Imagen	Descripción
negx.png	A square image showing a bright blue sky with white clouds and a dark blue ocean below.	Parte izquierda del fondo
negy.png	A square image showing a dark blue gradient, representing the bottom of the sky and sea.	Parte inferior del fondo
negz.png	A square image showing a bright blue sky with white clouds and a dark blue ocean below, similar to the left side image.	Parte trasera del fondo

posx.png		Parte derecha del fondo
posy.png		Parte superior del fondo
posz.png		Parte frontal del fondo

Bibliotecas

Como se mencionó en el análisis, se ocuparon las bibliotecas GLFW, GLM, GLAD y ASSIMP. Dichas bibliotecas se encuentran en la carpeta `deps` en el directorio con su respectivo nombre.

- ❖ GLFW nos permite trabajar con las ventanas.
- ❖ GLM es una biblioteca que nos permite realizar operaciones matemáticas de manera rápida.
- ❖ GLAD nos permite comunicarnos con la GPU, integra OpenGL.
- ❖ ASSIMP nos permite importar escenas gráficas en una gran variedad de formatos.

Interacción

Teclas y acciones con las que podemos interactuar con el modelo

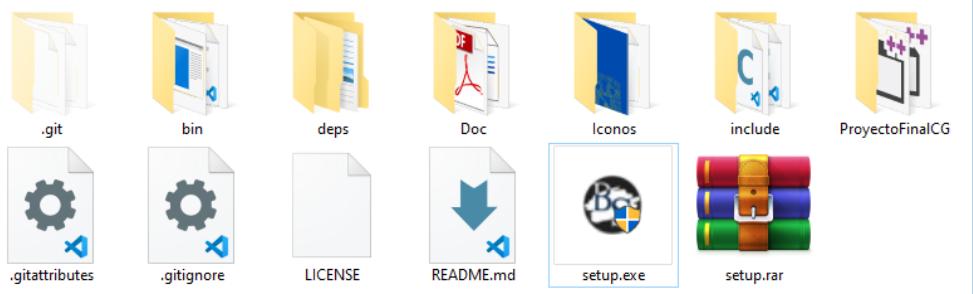
Tecla	Efecto sobre el modelo
W	Cámara se mueve hacia adelante
S	Cámara se mueve hacia atrás
A	Cámara se mueve a la derecha
D	Cámara se mueve a la izquierda
R	Aumenta velocidad cámara

T	Disminuye velocidad cámara
Key up	Personaje 1 se mueve hacia adelante
Key down	Personaje 1 se mueve hacia atrás
Key left	Rota el Personaje 1 a la derecha
Key right	Rota el Personaje 1 a la izquierda
I	Personaje 2 se mueve hacia adelante
K	Personaje 2 se mueve hacia atrás
J	Rota el Personaje 2 a la derecha
L	Rota el Personaje 2 a la izquierda
C	Prende la linterna
V	Apaga la linterna
+	Girar a la izquierda el Timón
-	Girar a la derecha el Timón
O	Aumenta el tamaño del timón
P	Disminuye el tamaño del timón
Z	Aumenta velocidad personaje 1
X	Disminuye velocidad personaje 1
N	Aumenta velocidad personaje 2
M	Disminuye velocidad personaje 2
F1	Cambia la animación del Personaje 1
F2	Cambia la animación del Personaje 1
F3	Cambia la animación del personaje 2
F4	Cambia la animación del personaje 2
ESC	Cerrar ventana
SCROLL	Zoom in/out de la cámara

Desarrollo

Directorio

El proyecto descargado desde Github tiene una estructura seccionada por carpetas.



Podemos observar diferentes archivos y la carpeta pertenecientes a Github. También podemos observar el instalador listo para ejecutarse y su versión comprimida en formato RAR.

Dentro de las carpetas de desarrollo, contamos con *Doc*, la cual cuenta con la documentación del proyecto, así como con archivos de ayuda como el mapeo de teclas y el manual de usuario.

En la carpeta *ProjectFinalCG* encontramos los archivos del proyecto relacionados a Visual Studio 2019. Aquí encontramos el archivo principal del proyecto, *ProjectFinalCG.cpp*, y los archivos relacionados al proyecto en Visual Studio, así como la solución en Visual Studio.

La carpeta *bin* contiene los archivos relacionados a la compilación de herramientas como ASSIMP, **y la compilación del proyecto mismo**. También cuenta con las carpetas *Models* y *shaders*.

Archivos de cabeceras adicionales

La carpeta *include* contiene las clases utilizadas para el desarrollo del proyecto que sirven como archivos de cabecera (bibliotecas o headers). Aquí encontramos los siguientes archivos:

- ***DirectionalLight.h, SpotLight.h:*** Clases relacionadas a los 2 tipos de luces que se pueden implementar en el modelo. Estas clases contienen los atributos necesarios para pasar la información de manera más organizada a los shaders.
- ***camera.h:*** La clase camera realiza la generación y manipulación de la cámara sintáctica
- ***model.h:*** La clase model permite cargar un modelo de un archivo usando ASSIMP.
- ***mesh.h:*** La clase mesh permite generar una malla de vértices para dibujarla.
- ***Material.h:*** La clase material contiene los atributos necesarios para definir las componentes de iluminación para un material.

- ***shader_m.h***: La clase shader es nuestro proxy entre los shaders compilados en la GPU y nuestro programa en C++.
- ***stb_image.h***: La clase stb_image nos ayuda a la carga de imágenes usando ASSIMP.

Shaders

Dentro de la carpeta `bin`, encontramos también la carpeta `shaders`. Dentro de esta carpeta encontramos seis diferentes conjuntos de shaders. Los shaders de vértices no cambian totalmente, la gran mayoría de ellos tiene la misma estructura, pero se decidió separarlos para realizar las pruebas de manera más organizada.

Shader	Modelo de iluminación	Descripción
fondo.vs		Contiene instrucciones básicas para el dibujo de la malla.
fondo.fs	Phong	Contempla el uso de 6 luces direccionales para iluminar el fondo hecho con la técnica skybox
objetosConTexturaVariasLuces.vs		Contiene instrucciones básicas para el dibujo de la malla.
objetosConTexturaVariasLuces.fs	Phong	Contempla el uso de texturas para los objetos, 5 luces direccionales y una spotlight para los materiales opacos que cuentan con texturas.
objetosSinTexturaVariasLuces.vs		Contiene instrucciones básicas para el dibujo de la malla.
objetosSinTexturaVariasLuces.fs	Phong	Contempla el uso de 5 luces direccionales y una spotlight sin textura para los materiales opacos como maderas, paredes y plásticos.
ObjetosSinTextura1Luz.vs		Contiene instrucciones básicas para el dibujo de la malla.
ObjetosSinTextura1Luz.fs	Phong	Contempla solo el uso de una luz direccional y una spotlight para materiales reflectantes como metales opacos.
animados.vs		Contiene instrucciones necesarias para la correcta interpretación de las armaduras para los personajes animados.
animados.fs	Phong	Contempla el uso de 5 luces

		direccionales y una spotlight sin textura para los materiales opacos como maderas, paredes y plásticos.
11_Fresnel.vs		Proporcionado por el profesor de laboratorio, realiza las operaciones necesarias para encontrar la reflexión y refracción para el modelo de Fresnel.
11_Fresnel.fs	Fresnel	Contiene las instrucciones necesarias para proveer a ciertos objetos como el vidrio y el oro con un efecto de reflexión.

Código principal

Diccionario de datos

Nombre	Tipo	Descripción
SCR_WIDTH	unsigned int	Define el ancho de la ventana.
SCR_HEIGHT	unsigned int	Define el alto de la ventana.
camera	Camera	Estructura de datos que define a la cámara sintética .
projection	glm::mat4	Matriz que define la matriz de proyección.
view	glm::mat4	Matriz que define la matriz de puerto de vista.
lastX	float	Indica la posición en X del ratón en el último movimiento de este.
lastY	float	Indica la posición en Y del ratón en el último movimiento de este.
firstMouse	bool	Bandera que indica si el ratón se ha movido anteriormente o no.
sol, posx, negx, posy, negy, posz, negz	DirectionalLight	Estructuras de datos que definen las propiedades para implementar una fuente de luz direccional.
Lintern	SpotLight	Estructura de datos que define las propiedades para implementar una luz de tipo spotlight, nos sirve para controlar las propiedades de la linterna.
window	GLFWwindow*	Apuntador para la ventana creada con GLFW.
bgR, bgG, bgB, bgA	GLfloat	Definen las componentes RGBA del fondo de la ventana.
Todos los shaders	Shade*	Apuntadores hacia los 6 diferentes

		tipos de shaders que se crean, compilan y cargan para la GPU.
Todos los modelos	Model*	Apuntadores de todos los modelos cargados.
ZoroBones	glm::mat4	Matriz para cargar los pesos de los huesos del personaje Zoro.
RobinBones	glm::mat4	Matriz para cargar los pesos de los huesos del personaje Robin.
fps	float	Indica la cantidad de fps a la que se mueven los personajes.
keys	int	Indica el número de frames en los que se mueven los personajes.
animationCount	int	Indica que pose dentro del arreglo de keyframes deben tener los personajes.
animationTypeZoro	int	Indica que animación se mostrará del personaje Zoro, la 0 o 1.
animationTypeRobin	int	Indica que animación se mostrará del personaje Robin, la 0 o 1.
deltaTime	float	Diferencia de tiempo entre el último frame y el actual
lastFrame	float	Tiempo en el que se ejecutó el anterior frame.
elapsedTime	float	Tiempo transcurrido de animaciones.
positionZoro	glm::vec3	Vector de posición del personaje Zoro.
forwardViewZoro	glm::vec3	Vector de vista del personaje Zoro.
scaleVZoro	float	Escala de velocidad del movimiento del personaje Zoro.
rotateCharacterZoro	float	Ángulo de rotación del personaje Zoro.
positionRobin	glm::vec3	Vector de posición del personaje Robin.
forwardViewRobin	glm::vec3	Vector de vista del personaje Robin.
scaleVRobin	float	Escala de velocidad del movimiento del personaje Robin.
rotateCharacterRobin	float	Ángulo de rotación del personaje Robin
timonRotation	float	Ángulo de rotación del timón.
escalaTimon	float	Factor de escalamiento del timón.
variables model	glm::mat4	Definen la matriz de modelo de las mallas que se dibujan.

Todos los materiales	Material	Estructura de datos que contiene las componentes necesarias para definir el material de un modelo. Hay uno por cada modelo.
modelTimon	glm::mat4	Matriz de modelo para el timón.
faces	vector<std::string>	Define las direcciones de las piezas que componen el skybox.
cubemapTexture	unsigned int	Índice que hace referencia a la carga de las imágenes que son las piezas del skybox.
modelZoro	glm::mat4	Matriz de modelo del personaje Zoro.
modelRobin	glm::mat4	Matriz de modelo del personaje Robin.
currentFrame	float	Tiempo actual.

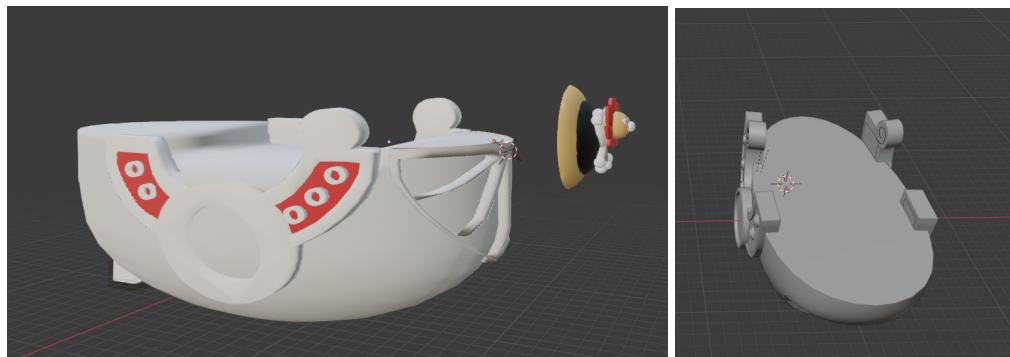
Diccionario de Funciones

Nombre	Parámetros	Tipo	Descripción
main	void	void	Función principal.
Start	void	bool	Inicializa la pantalla con GLFW, asigna funciones para tratar los callbacks, inicializa los shaders y modelos, determina la posición inicial de las animaciones.
InicializacionShaders	void	bool	Carga y compila los archivos de shaders, comprobando que se hayan compilado de manera correcta.
InicializacionModelos	void	bool	Carga los modelos y la información relevante de sus archivos FBX como texturas y huesos en caso de las animaciones.
MaterialesSinTextura UnaLuz	void	void	Dibuja los materiales sin textura con una luz direccional. Iluminación de Phong.
MaterialesSinTextura VariasLuces	void	void	Dibuja los materiales sin textura con varias luces direccionales. Iluminación de Phong.
MaterialesSinTextura Cocina	void	void	Dibuja los materiales sin textura con varias luces

			direccionales. Objetos de la cocina. Iluminación de Phong.
MaterialesConTexturasVariasLuces	void	void	Dibuja los materiales con textura con varias luces direccionales. Iluminación de Phong.
DibujaFondo	void	void	Dibuja el Fondo. Iluminación de Phong.
MaterialesSinTexturaFresnel	void	void	Dibuja los materiales sin textura con la iluminación de Fresnel.
MaterialesAnimadosVariasLuces	void	void	Dibuja los personajes con sus animaciones (poses). Iluminación de Phong.
Update	void	bool	Llama a las funciones anteriores para dibujar los modelos. Modifica valores útiles para las animaciones, el procesamiento de callbacks, luces, proyección y vista.
Limpiar	void	bool	Libera el espacio de memoria utilizado en la carga de modelos y de shaders.
processInput	GLFWwindow* window	void	Procesa las entradas de teclado.
framebuffer_size_callback	GLFWwindow* window, int width, int height	void	Procesa los cambios de tamaño de la ventana, específicamente el puerto de vista.
mouse_callback	GLFWwindow* window, double xpos, double ypos	void	Procesa las entradas de ratón (solo movimientos).
scroll_callback	GLFWwindow* window, double xoffset, double yoffset	void	Procesa el movimiento del ratón (scroll)
loadCubemap	vector<std::string> faces	unsigned int	Función que nos permite cargar las imágenes separadas del skybox para la iluminación de Fresnel.

Experimentos

Los primeros experimentos realizados fueron enfocados al modelado de las mallas. Utilizamos el software de Blender, para lo cual tuvimos que estudiar y tomar un curso de Blender en la plataforma de Youtube. Este aprendizaje temprano nos permitió ahorrar mucho tiempo al realizar los modelos del barco, ya que este se compone de formas complejas muy diferentes a un simple cubo y esfera.

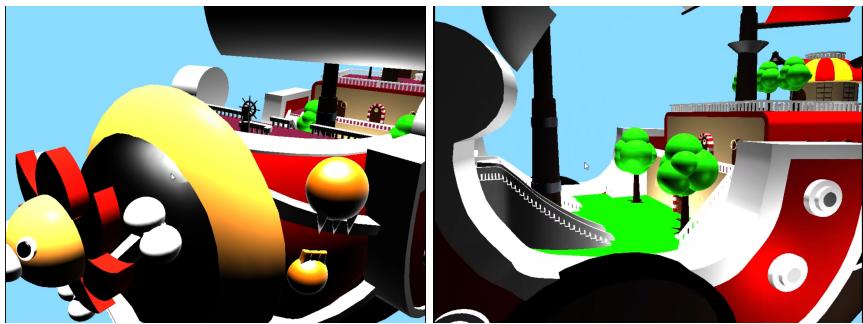


Modelado antes y después de tomar el curso de Blender (izquierda antes, derecha después)

La parte del barco fue realizándose tomando la parte trasera como base para poder darle buenas dimensiones al resto del barco. Podemos observar el proceso de modelado en las siguientes imágenes..



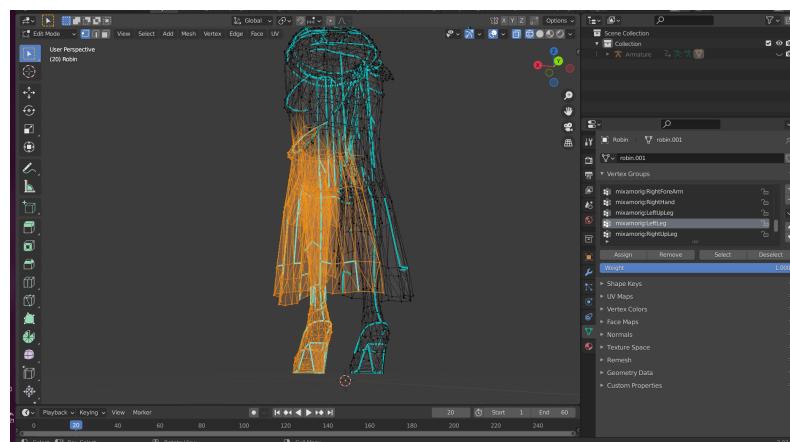
Durante la carga de modelos, se utilizó una sola luz direccional, la cual dio como resultado un modelo oscuro y con sombras muy intensas que no buscábamos tener en el modelo final. En las siguientes imágenes podemos observar los resultados de implementar solo una luz direccional y una luz spotlight.



Para tratar de corregir esto, se implementaron 4 luces puntuales al centro, costados y parte trasera del barco, pero la iluminación generada con estos 4 point lights no fue como se esperaba ya que las zonas más cercanas a la luz se iluminaban con mucha mayor intensidad, y las zonas más alejadas permanecían con las mismas sombras.

Por ello, se implementó el uso de 6 luces direccionales para poder iluminar en su totalidad el modelo. Algo que notamos fue que materiales muy reflejantes como los metales aparecían con muchos brillos especulares, resultado de las 6 luces, y otros elementos como los árboles y la madera blanca perdía su forma al no tener sombras que delimitaran sus caras. Dado esto, se dividieron los materiales entre los que ocuparían solo una luz direccional y los que ocuparían varias luces. El máximo de luces posicionales asignadas al final fueron 6 para el fondo, 5 para la mayoría de los materiales opacos y una sola luz posicional para los árboles y los metales como las cadenas. El resultado de combinar los diferentes tipos de luces los podemos ver en la parte de resultados.

Para la animación de los modelos, se utilizaron diferentes animaciones predefinidas en la plataforma Mixamo. Para el personaje 1, Zoro, no hubo ningún inconveniente al cargar el modelo y asignarle ciertas animaciones. No obstante, para el segundo personaje, Robin, hubo más problemas dada la falda que utiliza el personaje. El software de Mixamo no reconocía de manera correcta sus pies, por lo cual se tuvo que modificar los pesos de los huesos de los pies para que tuviera una correcta animación. Podemos observar la modificación a los pesos de la armadura en la siguiente imagen.



Link Video demostrativo

<https://youtu.be/1RQnYJANqV8>

Link proyecto en github

<https://github.com/AbeJLazaro/ThousandSunnyOpenGL>