



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Laboratorio de Computación Gráfica e Interacción

Humano-Computadora

**Proyecto Final
Manual técnico**

**Integrantes:
Mireles Estrada Daniel
Zavala Valdez Daniela Susana**

**Grupo de teoría: 2
Profesor: Carlos Aldair Roman Balbuena
Grupo de laboratorio: 9**

Semestre 2021-1

Fecha: 20/01/2021

Manual técnico

Definición y objetivo general del proyecto	2
Especificación de requerimientos del proyecto	2
Procedimientos de instalación y prueba	3
Alcance del proyecto	4
Diagrama de Gantt	4
Arquitectura del sistema	5
Construcción del proyecto	7
- Modelado y texturizado	
- Proceso de animación	
- Sonido	
- Documentación del código	
Contacto	16

Definición y objetivo general del proyecto

El proyecto consiste en un entorno virtual representando la casa de Los Simpson, serie norteamericana creada en 1987, el cual pretende permitirle al usuario conocer más de cerca esa famosa casa gracias a un recorrido virtual por cada una de las habitaciones de los personajes así como salas, cocina, cuarto de juegos, cocina y comedor; de igual manera, se puede interactuar con algunos elementos allí representados.

Especificación de requerimientos del proyecto

Este entorno fue creado desde cero, desde montar la idea de que era lo que se quería lograr, hasta decidir cuáles objetos se comienzan a modelar primero y el detalle que se les daría. Los modelos de 4 de los 5 personajes principales se descargaron de Internet por cuestión de tiempo, sin embargo, las animaciones se hicieron al gusto del programador.

Dicho entorno es capaz de hacer el recorrido por cada uno de los cuartos, las limitaciones que este recorrido presenta es que no se cuenta con todos los objetos de la casa (con la mayoría y los más relevantes sí), ni con todas las habitaciones de la misma de tal manera que falta el sótano y el ático, así mismo los personajes no tienen la capacidad de hacer el recorrido junto con el usuario.

Para poder desarrollarlo fue necesario utilizar softwares tales como: Visual Studio versión 2019, GIMP para la edición de texturas de los objetos y Blender para el modelado y creación de las animaciones.

Procedimientos de instalación y prueba

Creación del ejecutable

- Abrir el archivo “MED_ZVDS_ProyectoFinal_Gpo9.sln” desde el explorador de Windows o cargarlo desde Visual Studio 2019.
- Desde el menú Debug, seleccionar la opción Release y después iniciar el Depurador local de Windows como se observa en la Figura 1.

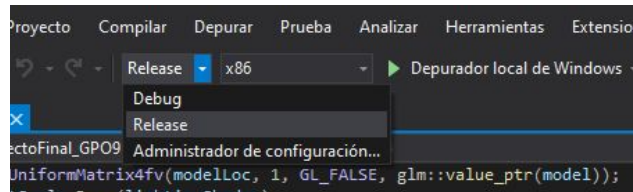


Figura 1. Opción Release y Botón Depurador local de Windows

- El programa debe comenzar a ejecutarse. Dependiendo de la computadora, puede tardar entre 1 a 5 minutos en cargar.
- Una vez cargado el proyecto debe visualizarse como se muestra en la Figura 2.

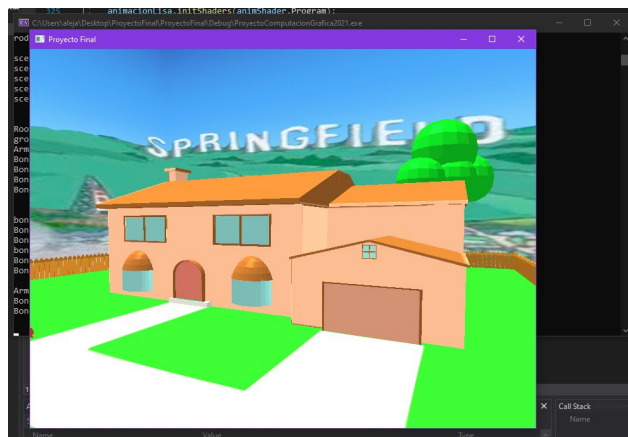


Figura 2. Proyecto funcionando

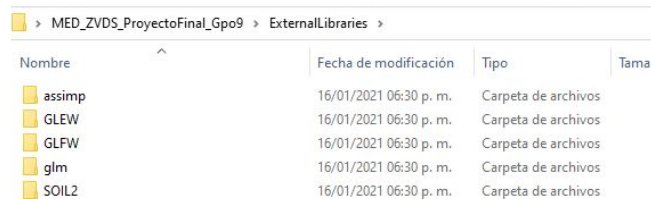
- Inmediatamente, en la misma donde se encuentra el archivo .sln, se crea una carpeta llamada Release, donde ya se está creado el ejecutable, sin embargo, es necesario cargar estos archivos de la carpeta donde se encuentra el proyecto completo para la correcta ejecución del mismo.

El proyecto se desarrolló a lo largo de 13 semanas.

Arquitectura del sistema

Carpetas y contenido

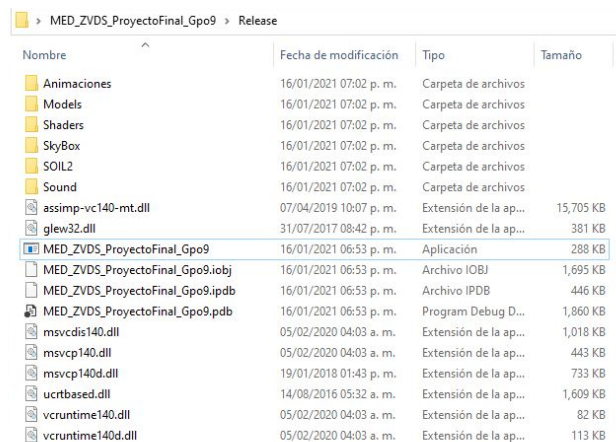
- ExternalLibraries: Contiene todas las bibliotecas necesarias para la creación del proyecto, así como sus funcionalidades.



Nombre	Fecha de modificación	Tipo	Tamaño
assimp	16/01/2021 06:30 p. m.	Carpeta de archivos	
GLEW	16/01/2021 06:30 p. m.	Carpeta de archivos	
GLFW	16/01/2021 06:30 p. m.	Carpeta de archivos	
glm	16/01/2021 06:30 p. m.	Carpeta de archivos	
SOIL2	16/01/2021 06:30 p. m.	Carpeta de archivos	

Figura 5. Contenido carpeta ExternalLibraries

- Release: Contiene los archivos necesarios para lanzar el ejecutable del proyecto mismos que son obtenidos de la carpeta principal



Nombre	Fecha de modificación	Tipo	Tamaño
Animaciones	16/01/2021 07:02 p. m.	Carpeta de archivos	
Models	16/01/2021 07:02 p. m.	Carpeta de archivos	
Shaders	16/01/2021 07:02 p. m.	Carpeta de archivos	
SkyBox	16/01/2021 07:02 p. m.	Carpeta de archivos	
SOIL2	16/01/2021 07:02 p. m.	Carpeta de archivos	
Sound	16/01/2021 07:02 p. m.	Carpeta de archivos	
assimp-vc140-mt.dll	07/04/2019 10:07 p. m.	Extensión de la ap...	15,705 KB
glew32.dll	31/07/2017 08:42 p. m.	Extensión de la ap...	381 KB
MED_ZVDS_ProyectoFinal_Gpo9	16/01/2021 06:53 p. m.	Aplicación	288 KB
MED_ZVDS_ProyectoFinal_Gpo9.iobj	16/01/2021 06:53 p. m.	Archivo IOBJ	1,695 KB
MED_ZVDS_ProyectoFinal_Gpo9.ipdb	16/01/2021 06:53 p. m.	Archivo IPDB	446 KB
MED_ZVDS_ProyectoFinal_Gpo9.pdb	16/01/2021 06:53 p. m.	Program Debug D...	1,860 KB
msvcds140.dll	05/02/2020 04:03 a. m.	Extensión de la ap...	1,018 KB
msvcp140.dll	05/02/2020 04:03 a. m.	Extensión de la ap...	443 KB
msvcp140d.dll	19/01/2018 01:43 p. m.	Extensión de la ap...	733 KB
ucrtbased.dll	14/08/2016 05:32 a. m.	Extensión de la ap...	1,609 KB
vcruntime140.dll	05/02/2020 04:03 a. m.	Extensión de la ap...	82 KB
vcruntime140d.dll	05/02/2020 04:03 a. m.	Extensión de la ap...	113 KB

Figura 6. Contenido carpeta Release

- Carpeta del proyecto completo

MED_ZVDS_ProyectoFinal_Gpo9 > MED_ZVDS_ProyectoFinal_Gpo9 >			
Nombre	Fecha de modificación	Tipo	Tamaño
Animaciones	16/01/2021 06:29 p. m.	Carpeta de archivos	
Debug	16/01/2021 06:47 p. m.	Carpeta de archivos	
Models	16/01/2021 06:29 p. m.	Carpeta de archivos	
Release	16/01/2021 06:53 p. m.	Carpeta de archivos	
Shaders	16/01/2021 06:29 p. m.	Carpeta de archivos	
SkyBox	16/01/2021 06:29 p. m.	Carpeta de archivos	
SOIL2	16/01/2021 06:29 p. m.	Carpeta de archivos	
Sound	16/01/2021 06:29 p. m.	Carpeta de archivos	
assimp-vc140-mt.dll	07/04/2019 10:07 p. m.	Extensión de la ap...	15,705 KB
Camera.h	12/01/2021 10:53 p. m.	C/C++ Header	5 KB
glew32.dll	31/07/2017 08:42 p. m.	Extensión de la ap...	381 KB
*+ main_proyecto_final.cpp	14/01/2021 05:27 p. m.	C++ Source	61 KB
MED_ZVDS_ProyectoFinal_Gpo9.vcxproj	16/01/2021 06:47 p. m.	VC++ Project	10 KB
MED_ZVDS_ProyectoFinal_Gpo9.vcxproj.f...	16/01/2021 06:47 p. m.	VC++ Project Filte...	3 KB
MED_ZVDS_ProyectoFinal_Gpo9.vcxproj....	16/01/2021 06:27 p. m.	Per-User Project O...	1 KB
Mesh.h	22/12/2020 10:59 p. m.	C/C++ Header	5 KB
meshAnim.h	22/12/2020 10:58 p. m.	C/C++ Header	7 KB
Model.h	22/12/2020 10:57 p. m.	C/C++ Header	10 KB
modelAnim.h	23/12/2020 01:23 a. m.	C/C++ Header	27 KB
Shader.h	25/03/2019 03:38 a. m.	C/C++ Header	4 KB
stb_image.h	09/01/2019 06:03 a. m.	C/C++ Header	249 KB
Texture.h	28/04/2019 11:47 p. m.	C/C++ Header	3 KB

Figura 7. Contenido carpeta principal

Dentro de esta última carpeta se encuentran:

- Los modelos animados en formato fbx exportados de Blender donde se creó la animación. (Animaciones)
- Los modelos que componen cada mueble de la casa así como la fachada en formato obj para ser importado en OpenGL con su respectiva imagen para la textura con formato png y su mtl. (Models)
- Cada uno de los shaders empleados para la representación gráfica. (Shaders)
- Las imágenes que componen el cubo para el Skybox en formato jpg. (Skybox)
- Los audios que componen parte de la interacción con el usuario. (Sound)
- Cada uno de los headers empleados en el proyecto así como sus respectivos dll.

Dependencias

```

c++ - General
$(SolutionDir)/External Libraries/GLEW/include
$(SolutionDir)/External Libraries/GLFW/include
$(SolutionDir)/External Libraries/glm
$(SolutionDir)/External Libraries/assimp/include

Linker - General
$(SolutionDir)/External Libraries/GLEW/lib/Release/Win32
$(SolutionDir)/External Libraries/GLFW/lib-vc2015
$(SolutionDir)/External Libraries/SOIL2/lib
$(SolutionDir)/External Libraries/glm
$(SolutionDir)/External Libraries/assimp/lib

Linker - Entrada
assimp-vc140-mt.lib;opengl32.lib;glew32.lib;glfw3.lib;soil2-debug.lib;

```

- De esta manera es cómo se agregan las bibliotecas en las Propiedades del proyecto dentro de VisualStudio.

Módulos cargados en VisualStudio

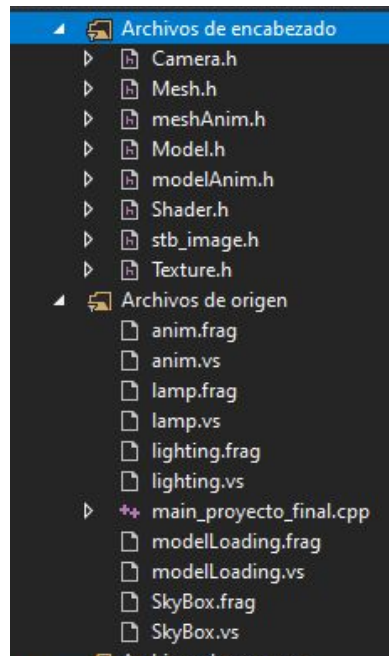


Figura 8. Carga de .h y shaders

- Mientras que los headers y los shaders se cargan como archivos de encabezado y archivos de origen, correspondientemente.

Construcción del proyecto

Modelado y texturizado

Todo el modelado se realizó a partir de Mesh básicos en Blender como los son cubos, cilindros y esferas, así como se hizo uso de las transformaciones como escala, rotación y traslación para imitar los muebles y fachada. Además, si era necesario, se utilizaron Scenes cuando un mueble tenía varias partes u otros objetos y así, unirlos.

Por parte del texturizado, en Blender se selecciona todo el objeto ya construido, con la combinación CTRL+T en modo edición, se triangulan las caras para su carga en OpenGL, además de construir el MapaUV. A partir de dicho mapa, se carga en GIMP para realizar la textura y con la edición de imágenes por capas en este software se comienza a pintar la capa de los colores, dejando intacto el UV. Regresando a Blender, en la pestaña de Shading, se asigna el material, la textura y se carga la nueva imagen.

Proceso de animación

Las animaciones se hicieron con ayuda de los huesitos y keyframes. El proceso de skinning y rigging fue el siguiente:

- Ya con nuestro modelo texturizado dentro del software de modelado Blender se creó una armadura. Ésta nos permite crear un huesito y posteriormente ir creando los necesarios para lograr la articulación deseada.
- Debido a problemas con la asignación de las armaduras se hizo uso de la herramienta fusionar por distancia (Merge by distance), la cual nos permite ir fusionando los vértices deseados y que estén más cercanos.
- Blender asigna las prioridades de los huesitos de forma automática y esto trae como consecuencia la deformación del modelo al momento de hacer la animación, por lo tanto, se usa la opción de Weight Paint, está de permite indicarle al software que parte del modelo es la que se desea mover al hacer el movimiento de un hueso en particular (Esta parte del proceso requiere de mucho tiempo por todas las pruebas que se hacen para verificar que el movimiento es el deseado).
- Ya que tenemos estos movimientos insertamos los keyframes. Para esto se debe guardar las transformaciones ya hechas para cada uno de los huesos.
- Ya en este parte podemos hacer más rápido el movimiento de la animación haciendo más pequeña la distancia entre los keyframes.
- Como paso final e importante se debe de reiniciar la posición, rotación y escala de la armadura para que la exportación del modelo se haga correctamente.

Sonido

Para cargar el sonido en el código se utiliza el API de Windows que tiene métodos para reproducir y detener sonidos. Tiene algunas limitaciones pues no se logra reproducir varios sonidos a la vez, sin embargo, su implementación es sencilla.

Solamente se manda a llamar en el código Windows.h, se añade el header y se agrega la acción de las teclas, asignada a la ruta del audio que se desea reproducir, para controlar esa interacción.

Documentación del código

De la línea 148 a la 314 se hizo la declaración de los modelos de cada uno de los objetos tanto del exterior como del interior de la casa.


```

//+++++Cargar los modelos+++++//
//+++++//
//EN ESTA PARTE DEL CODIGO SE LE DA UN NOMBRE A NUESTRO MODELO Y MANDAMOS A LLAMAR NUESTRO ARCHIVO .OBJ
// CON SU RESPECTIVO MTL, DAMOS LA RUTA DE LA ANIDACION DE LAS CARPETAS
// SE HIZO LA SEPARACION DE LOS OBJETOS DE PLANTA ALTA Y PLANTA BAJA SEPARANDOLOS POR CUARTOS

/*Para los modelos de la fachada, las separaciones de los cuartos y sus ventanas*/
Model ourModelFachada((char*)"Models/CasaSimpson/Fachada.obj");
Model ourModelSegundoPiso((char*)"Models/CasaSimpson/SegundoPiso.obj");
Model ourModelPrimerPiso((char*)"Models/CasaSimpson/PrimerPiso.obj");
Model ExtVentanas((char*)"Models/CasaSimpson/VENTANAS/VENTANASFRENTE/VentanasFrente.obj");
Model IntVentanas((char*)"Models/CasaSimpson/VENTANAS/VENTANASREGULARES/VentanasRegular.obj");

//-----Fuera de la casa -----//
/*Se mandan a llamar los objetos de la parte exterior de la casa*/
Model ExtArbol((char*)"Models/CasaSimpson/Fuera/Arbol1.obj");
Model ExtArbusto((char*)"Models/CasaSimpson/Fuera/Arbusto.obj");
Model ExtBarda((char*)"Models/CasaSimpson/Fuera/Barda.obj");
Model ExtBuzon((char*)"Models/CasaSimpson/Fuera/Buzon.obj");
Model ExtArena((char*)"Models/CasaSimpson/Fuera/CajaArena.obj");
Model ExtCasaArbol((char*)"Models/CasaSimpson/Fuera/CasaArbol.obj");
Model ExtPerro((char*)"Models/CasaSimpson/Fuera/CasaPerro.obj");
Model ExtManguera((char*)"Models/CasaSimpson/Fuera/Manguera.obj");
Model ExtSuelo((char*)"Models/CasaSimpson/Fuera/Suelo.obj");

```

Figura 9.Declaración de modelos planta baja

```

/*****SALA PIANO*****/
Model CPBanco((char*)"Models/CasaSimpson/CUARTO PIANO/Banco.obj");
Model CPBurosPiano((char*)"Models/CasaSimpson/CUARTO PIANO/BurosPiano.obj");
Model CPChimenea((char*)"Models/CasaSimpson/CUARTO PIANO/Chimenea.obj");
Model CPLamparasDeMesaPiano((char*)"Models/CasaSimpson/CUARTO PIANO/LamparasDeMesaPiano.obj");
Model CPMasaConPlato((char*)"Models/CasaSimpson/CUARTO PIANO/MasaConPlato.obj");
Model CPMesaCentro((char*)"Models/CasaSimpson/CUARTO PIANO/MesaCentro.obj");
Model CPMesaConLampara((char*)"Models/CasaSimpson/CUARTO PIANO/MesaConLampara.obj");
Model CPMuebleChungo((char*)"Models/CasaSimpson/CUARTO PIANO/MuebleChungo.obj");
Model CPPiano((char*)"Models/CasaSimpson/CUARTO PIANO/Piano.obj");
Model CPSillaSalaPiano((char*)"Models/CasaSimpson/CUARTO PIANO/SillaSalaPiano.obj");
Model CPSillonChico((char*)"Models/CasaSimpson/CUARTO PIANO/SillonChico.obj");
Model CPSillonGrande((char*)"Models/CasaSimpson/CUARTO PIANO/SillonGrande.obj");
Model CPSillonTazones((char*)"Models/CasaSimpson/CUARTO PIANO/Tazones.obj");
Model CPSillonTrofeo1((char*)"Models/CasaSimpson/CUARTO PIANO/Trofeo1.obj");
Model CPSillonTrofeo3((char*)"Models/CasaSimpson/CUARTO PIANO/Trofeo3.obj");
Model CPSillonTrofeo2((char*)"Models/CasaSimpson/CUARTO PIANO/Trofeo2.obj");
Model CPSillonTrofeos((char*)"Models/CasaSimpson/CUARTO PIANO/Trofeos.obj");

/*****SALA TV*****/
Model STburo((char*)"Models/CasaSimpson/SALA TV/buro.obj");
Model STcuadropared((char*)"Models/CasaSimpson/SALA TV/cuadropared.obj");
Model STLamparaMesa((char*)"Models/CasaSimpson/SALA TV/lamparaMesa.obj");
Model STLamparaPiso((char*)"Models/CasaSimpson/SALA TV/lamparaPiso.obj");
Model STphone((char*)"Models/CasaSimpson/SALA TV/phone.obj");
Model STsillon((char*)"Models/CasaSimpson/SALA TV/sillon.obj");
Model STstereo((char*)"Models/CasaSimpson/SALA TV/stereo.obj");
Model STtv((char*)"Models/CasaSimpson/SALA TV/tv.obj");

```

Figura 10.Declaración de modelos planta baja


```
//-----Cuarto de juegos-----//
Model JEstante((char*)"Models/CasaSimpson/CUARTOJUEGOS/estante/Estante.obj");
Model JMesaDeTV((char*)"Models/CasaSimpson/CUARTOJUEGOS/mesa_de_tv/MesaDeTv.obj");
Model JMuebleJarrones((char*)"Models/CasaSimpson/CUARTOJUEGOS/mueble_con_jarrones/MuebleJarrones.obj");
Model JSillonJuegos((char*)"Models/CasaSimpson/CUARTOJUEGOS/sillon/Sillon.obj");
Model Jvmaggie((char*)"Models/CasaSimpson/CUARTOJUEGOS/tvmaggie/Tv.obj");
Model JCajonChico1((char*)"Models/CasaSimpson/CUARTOJUEGOS/cajonparedchico/CajonChico1.obj");
Model JCajonChico2((char*)"Models/CasaSimpson/CUARTOJUEGOS/cajonparedchico/CajonChico2.obj");
Model JCajonChico3((char*)"Models/CasaSimpson/CUARTOJUEGOS/cajonparedchico/CajonChico3.obj");
Model JCajonChico4((char*)"Models/CasaSimpson/CUARTOJUEGOS/cajonparedchico/CajonChico4.obj");
Model JCajonChico5((char*)"Models/CasaSimpson/CUARTOJUEGOS/cajonparedchico/CajonChico5.obj");
Model JCajonChico6((char*)"Models/CasaSimpson/CUARTOJUEGOS/cajonparedchico/CajonChico6.obj");
Model JCajonChico7((char*)"Models/CasaSimpson/CUARTOJUEGOS/cajonparedchico/CajonChico7.obj");
Model JCajonChico8((char*)"Models/CasaSimpson/CUARTOJUEGOS/cajonparedchico/CajonChico8.obj");
Model JCajonGrande1((char*)"Models/CasaSimpson/CUARTOJUEGOS/cajonparedgrande/CajonGrande1.obj");
Model JCajonGrande2((char*)"Models/CasaSimpson/CUARTOJUEGOS/cajonparedgrande/CajonGrande2.obj");

//-----Pasillo Planta baja-----//

Model PEscritorioPasillo((char*)"Models/CasaSimpson/PASILLOPB/Escritorio/Escritorio.obj");
Model PReloj((char*)"Models/CasaSimpson/PASILLOPB/Reloj/Reloj.obj");

//-----Comedor-----//

Model ComeBufetera((char*)"Models/CasaSimpson/COMEDOR/bufetera/bufetera.obj");
Model ComeCuadro((char*)"Models/CasaSimpson/COMEDOR/cuadro/cuadro.obj");
Model ComeMesa((char*)"Models/CasaSimpson/COMEDOR/mesa/mesa.obj");
Model ComeSilla((char*)"Models/CasaSimpson/COMEDOR/silla/silla.obj");
```

Figura 11. Declaración de modelos planta baja

```
//+++++PLANTA BAJA+++++//
//+++++//

/*****COCINA*****/
Model CocinaCenArribaDer((char*)"Models/CasaSimpson/COCINA/cenArribaDer.obj");
Model CocinaCenArribaIzq((char*)"Models/CasaSimpson/COCINA/cenArribaIzq.obj");
Model CocinaDerAbajo((char*)"Models/CasaSimpson/COCINA/derAbajo.obj");
Model CocinaDerArriba((char*)"Models/CasaSimpson/COCINA/derArriba.obj");
Model CocinaEstufa((char*)"Models/CasaSimpson/COCINA/estufa.obj");
Model CocinaFregadero((char*)"Models/CasaSimpson/COCINA/fregadero.obj");
Model CocinaIzqAbajo((char*)"Models/CasaSimpson/COCINA/izqAbajo.obj");
Model CocinaIzqArriba((char*)"Models/CasaSimpson/COCINA/izqArriba.obj");
Model Cocinalavaplatos((char*)"Models/CasaSimpson/COCINA/lavaplatos.obj");
Model Cocinalicuada((char*)"Models/CasaSimpson/COCINA/licuada.obj");
Model Cocinarefri((char*)"Models/CasaSimpson/COCINA/refri.obj");
Model CocinasillaC((char*)"Models/CasaSimpson/COCINA/sillaC.obj");
Model Cocinamesa((char*)"Models/CasaSimpson/COCINA/mesacocina.obj");

/*****COCHERA*****/
Model CocheraCaja((char*)"Models/CasaSimpson/COCHERA/caja.obj");
Model CocheraCajaAbierta((char*)"Models/CasaSimpson/COCHERA/CajaAbierta.obj");
Model CocheraEscobaMedieval((char*)"Models/CasaSimpson/COCHERA/EscobaMedieval.obj");
Model CocheraLatas((char*)"Models/CasaSimpson/COCHERA/Latas.obj");
Model CocheraLocker((char*)"Models/CasaSimpson/COCHERA/Locker.obj");
Model CocheraMesaCochera((char*)"Models/CasaSimpson/COCHERA/MesaCochera.obj");
Model CocheraTablaHerramientas((char*)"Models/CasaSimpson/COCHERA/TablaHerramientas.obj");
Model CocheraTablaHerramientasPared2((char*)"Models/CasaSimpson/COCHERA/TablaHerramientasPared2.obj");
```

Figura 12. Declaración de modelos planta baja


```
//+++++PLANTA ALTA+++++//
//+++++//

//-----Habitación de Bart-----//
Model BartCama((char*)"Models/CasaSimpson/BART/cama.obj");
Model BartBuroes((char*)"Models/CasaSimpson/BART/Buroes.obj");
Model BartLibrero((char*)"Models/CasaSimpson/BART/Librero.obj");
Model BartEscritorio((char*)"Models/CasaSimpson/BART/Escritorio.obj");
Model BartEspejo((char*)"Models/CasaSimpson/BART/Tocador.obj");
Model BartPizarron((char*)"Models/CasaSimpson/BART/Pizarron.obj");
Model BartRepisa((char*)"Models/CasaSimpson/BART/Repisa.obj");
Model BartSilla((char*)"Models/CasaSimpson/BART/Silla.obj");

//-----Habitación de Lisa-----//
Model LisaCama((char*)"Models/CasaSimpson/LISA/Cama.obj");
Model LisaCuadro((char*)"Models/CasaSimpson/LISA/Cuadro.obj");
Model LisaAtril((char*)"Models/CasaSimpson/LISA/Atril.obj");
Model LisaBuroes((char*)"Models/CasaSimpson/LISA/Buroes.obj");
Model LisaEspejo((char*)"Models/CasaSimpson/LISA/Espejo.obj");
Model LisaEstante((char*)"Models/CasaSimpson/LISA/Estante.obj");
Model LisaLampara((char*)"Models/CasaSimpson/LISA/Lampara.obj");
Model LisaLibrero((char*)"Models/CasaSimpson/LISA/Librero.obj");
Model LisaSilla((char*)"Models/CasaSimpson/LISA/Silla.obj");
```

Figura 13. Declaración de modelos planta alta

```
/******BAÑO HOMERO*****//
Model Ducha((char*)"Models/CasaSimpson/HOMERO-MARGE/BAÑO/Ducha.obj");
Model WC((char*)"Models/CasaSimpson/HOMERO-MARGE/BAÑO/WC.obj");
Model MueblePared((char*)"Models/CasaSimpson/HOMERO-MARGE/BAÑO/MueblePared.obj");
Model Mueble((char*)"Models/CasaSimpson/HOMERO-MARGE/BAÑO/Mueble.obj");
Model Lavavo((char*)"Models/CasaSimpson/HOMERO-MARGE/BAÑO/Lavavo.obj");
Model Regadera((char*)"Models/CasaSimpson/HOMERO-MARGE/BAÑO/Regadera.obj");
Model Espejo((char*)"Models/CasaSimpson/HOMERO-MARGE/BAÑO/Espejo.obj");
Model Toallero((char*)"Models/CasaSimpson/HOMERO-MARGE/BAÑO/Toallero.obj");

/******CUARTO HOMERO Y MARGE*****//

Model HMCama((char*)"Models/CasaSimpson/HOMERO-MARGE/Cama.obj");
Model HMBuroReloj((char*)"Models/CasaSimpson/HOMERO-MARGE/BuroReloj.obj");
Model HMBuroLampara((char*)"Models/CasaSimpson/HOMERO-MARGE/BuroLampara.obj");
Model HMTocador((char*)"Models/CasaSimpson/HOMERO-MARGE/Tocador.obj");
Model HMBanco((char*)"Models/CasaSimpson/HOMERO-MARGE/Banco.obj");
Model HMEspejo((char*)"Models/CasaSimpson/HOMERO-MARGE/Espejo.obj");
Model HMSillon((char*)"Models/CasaSimpson/HOMERO-MARGE/Sillon.obj");
Model HMComoda((char*)"Models/CasaSimpson/HOMERO-MARGE/Comoda.obj");
Model HMAlmohada((char*)"Models/CasaSimpson/HOMERO-MARGE/Almohada.obj");

/******MAGGIE*****//
Model MaggieCuna((char*)"Models/CasaSimpson/MAGGIE/Cuna.obj");
Model MaggieCorral((char*)"Models/CasaSimpson/MAGGIE/Corralito.obj");
Model MaggieJuguetero((char*)"Models/CasaSimpson/MAGGIE/Juguetero.obj");
Model MaggieBuro((char*)"Models/CasaSimpson/MAGGIE/Buro.obj");
Model MaggieColumpio((char*)"Models/CasaSimpson/MAGGIE/Columpio.obj");
Model MaggieCesto((char*)"Models/CasaSimpson/MAGGIE/Cesto.obj");
```

Figura 14. Declaración de modelos planta baja

```
//-----Baño Planta Alta-----//

Model BanoBote((char*)"Models/CasaSimpson/BañoPA/Bote.obj");
Model BanoEspejo((char*)"Models/CasaSimpson/BañoPA/Espejo.obj");
Model BanoEstante((char*)"Models/CasaSimpson/BañoPA/Estante.obj");
Model BanoLavabo((char*)"Models/CasaSimpson/BañoPA/Lavabo.obj");
Model BanoTina((char*)"Models/CasaSimpson/BañoPA/Tina.obj");
Model BanoWC((char*)"Models/CasaSimpson/BañoPA/WC.obj");
```

Figura 15. Declaración de modelos planta alta

De la línea 320 a la 337 se hace la declaración de los modelos de los personajes

```
// Cargar modelos de animación
//SE DEFINE EL NOMBRE DEL MODELO CON SU ANIMACION DE CADA UNO DE LOS PERSONAJES
// MANDANDO A LLAMAR SU RESPECTIVO ARCHIVO .FBX
ModelAnim animacionHomero("Animaciones/Homero/Homer_bones.fbx");
animacionHomero.initShaders(animShader.Program);

ModelAnim animacionBart("Animaciones/Bart/bart_bones.fbx");
animacionBart.initShaders(animShader.Program);

ModelAnim animacionLisa("Animaciones/Lisa/Lisa_bones.fbx");
animacionLisa.initShaders(animShader.Program);

ModelAnim animacionSnail("Animaciones/Snail/snail.fbx");
animacionSnail.initShaders(animShader.Program);

ModelAnim animacionMarge("Animaciones/Marge/Marge_Bones.fbx");
animacionMarge.initShaders(animShader.Program);

ModelAnim animacionToy("Animaciones/Toy_Ball/Toy_Ball.fbx");
animacionToy.initShaders(animShader.Program);
```

Figura 16 Declaración de modelos de los personajes

Indicamos cuales son los archivos que se cargarán en cada una de las caras de nuestro SkyBox, recordando que este está formado por un cubo.

```
// Cargar SKYBOX
// SE MANDA A LLAMAR CADA UNO DE LAS PARTES DE DICHO CUBO DESARMADO
// CON EL FIN DE QUE AL MOMENTO DE CARGARSE EL CUBO DE ARME
vector<const GLchar*> faces;
faces.push_back("SkyBox/Simpson/right.jpg");
faces.push_back("SkyBox/Simpson/left.jpg");
faces.push_back("SkyBox/Simpson/top.jpg");
faces.push_back("SkyBox/Simpson/bottom.jpg");
faces.push_back("SkyBox/Simpson/back.jpg");
faces.push_back("SkyBox/Simpson/front.jpg");
```

Figura 17. Declaración de las caras del SkyBox

Se definen los vértices para la creación de nuestro cubo para el SkyBox

```
// SE DEFINEN LOS VERTICES DEL CUBO CON EL CUAL ARMAMOS NUESTRO SKYBOX
GLfloat skyboxVertices[] = {
    // Positions
    -1.0f,  1.0f, -1.0f,
    -1.0f, -1.0f, -1.0f,
    1.0f,  -1.0f, -1.0f,
    1.0f, -1.0f, -1.0f,
    1.0f,  1.0f, -1.0f,
    -1.0f,  1.0f, -1.0f,

    -1.0f, -1.0f,  1.0f,
    -1.0f, -1.0f, -1.0f,
    -1.0f,  1.0f, -1.0f,
    -1.0f,  1.0f, -1.0f,
    -1.0f,  1.0f,  1.0f,
    -1.0f, -1.0f,  1.0f,

    1.0f, -1.0f, -1.0f,
    1.0f, -1.0f,  1.0f,
    1.0f,  1.0f,  1.0f,
    1.0f,  1.0f,  1.0f,
    1.0f,  1.0f, -1.0f,
    1.0f, -1.0f, -1.0f,

    -1.0f, -1.0f,  1.0f,
    -1.0f,  1.0f,  1.0f,
    1.0f,  1.0f,  1.0f,
    1.0f,  1.0f,  1.0f,
    1.0f, -1.0f,  1.0f,
    -1.0f, -1.0f,  1.0f,
}
```

Figura 18 Vértices para el cubo

De la línea 532 a la 1325 se mandan a dibujar cada uno de los modelos de los objetos ya previamente declarados.

```
//Carga de modelos
// EN ESTA PARTE DEL CODIGO SE MANDAN A DIBUJAR CADA UNO DE LOS MODELOS DE CADA OBJETO
-----Fachada y pisos -----

glm::mat4 model(1);

//Fachada
view = camera.GetViewMatrix();
//model = glm::translate(tmp, glm::vec3(-0.5f, 0.0f, -0.1f));
//model = glm::translate(model, glm::vec3(posX, posY, posZ));
//model = glm::rotate(model, glm::radians(rot), glm::vec3(0.0f, 1.0f, 0.0f));
//model = glm::rotate(model, glm::radians(-rotRodIzq), glm::vec3(1.0f, 0.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ourModelFachada.Draw(lightningShader);

//Segundo Piso
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ourModelSegundoPiso.Draw(lightningShader);

//Primer Piso
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ourModelPrimerPiso.Draw(lightningShader);

//Ventanas Exteriores
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ExtVentanas.Draw(lightningShader);
```

Figura 19. Se mandan a dibujar modelos de la parte exterior de la casa

```

//-----Fuera de la casa-----

//Arbol
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ExtArbol.Draw(lightingShader);

//Arbusto
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ExtArbusto.Draw(lightingShader);

//Barda
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ExtBarda.Draw(lightingShader);

//Buzon
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ExtBuzon.Draw(lightingShader);

```

Figura 20. Se mandan a dibujar modelos de la parte exterior de la casa

```

+++++PLANTA BAJA+++++
/*****COCINA*****/

//CenArribaDer
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
CocinaCenArribaDer.Draw(lightingShader);

//CenArribaIzq
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
CocinaCenArribaIzq.Draw(lightingShader);

//DerAbajo
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
CocinaDerAbajo.Draw(lightingShader);

//DerArriba
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
CocinaDerArriba.Draw(lightingShader);

```

Figura 21. Se mandan a dibujar modelos de la planta baja

```

+++++PLANTA ALTA+++++

-----Habitación de Bart-----//

//Cama
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
BartCama.Draw(lightningShader);

//Buroes
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
BartBuroes.Draw(lightningShader);

//Librero
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
BartLibrero.Draw(lightningShader);

//Escritorio
view = camera.GetViewMatrix();
//model = glm::translate(model, glm::vec3(0, -0.9f, -0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
BartEscritorio.Draw(lightningShader);

```

Figura 22. Se mandan a dibujar modelos de la planta alta

Para activar los efectos de sonido se utilizan las teclas del 1 al 5 y la tecla A, cada una activa un sonido diferente.

```

// TECLAS QUE SIRVEN PARA ACTIVAR EL SONIDO
if (keys[GLFW_KEY_1])
    PlaySound(TEXT("Sound/doh.wav"), NULL, SND_ASYNC | SND_FILENAME);

if (keys[GLFW_KEY_2])
    PlaySound(TEXT("Sound/burp.wav"), NULL, SND_ASYNC | SND_FILENAME);

if (keys[GLFW_KEY_3])
    PlaySound(TEXT("Sound/bartlaf.wav"), NULL, SND_ASYNC | SND_FILENAME);

if (keys[GLFW_KEY_4])
    PlaySound(TEXT("Sound/organized.wav"), NULL, SND_ASYNC | SND_FILENAME);

if (keys[GLFW_KEY_5])
    PlaySound(TEXT("Sound/Simpsons.wav"), NULL, SND_ASYNC | SND_FILENAME);

if (keys[GLFW_KEY_A])
    PlaySound(TEXT("Sound/hidden.wav"), NULL, SND_ASYNC | SND_FILENAME);

if (keys[GLFW_KEY_0])
    PlaySound(NULL, 0, 0);

```

Figura 22. Teclas para activar sonidos

Se le dio un punto inicial a la cámara para mejor la vista que tenemos de nuestro entorno

```
// Camera
/*SE MODIFICO LA VELOCIDAD DE LA CAMARA CON EL FIN DE HACER UN POCO MAS RAPIDO EL RECORRIDO*/
/*ESTE CAMBIO SE LLEVO A CABO EN EL ARCHIVO Camera.h*/

Camera camera(glm::vec3(650.0f, 20.0f, 0.0f));
GLfloat lastX = WIDTH / 2.0;
GLfloat lastY = HEIGHT / 2.0;
bool keys[1024];
bool firstMouse = true;
float range = 0.0f;
float rot = 0.0f;
```

Figura 23. Punto inicial de la cámara

Contacto

Para cualquier duda, comentario, sugerencia o problema que se presente durante el uso del sistema se puede comunicar a:

Mireles Estrada Daniel
danielmireles706@gmail.com
Zavala Valdez Daniela Susana
daniela.zavala107@gmail.com