

Proyecto Final: Recreación en 3D de la Casa de CJ, del juego Grand Theft Auto: San Andreas

Índice

MANUAL DE USUARIO

I.	INTRODUCCIÓN	3
II.	DESCARGA, COMPILACIÓN Y EJECUCIÓN	3
III.	INTERACCIONES	6

MANUAL TÉCNICO

I.	DESCRIPCIÓN DEL PROYECTO	9
II.	OBJETIVOS Y CONSIDERACIONES	10
III.	ALCANCE	11
IV.	LIMITANTES	12
V.	DIAGRAMA DE GANTT	12
VI.	REQUERIMIENTOS	13
VII.	DISEÑO DEL SISTEMA	14
VIII.	ESPECIFICACIONES FUNCIONALES	14
IX.	ANÁLISIS DE COSTOS	16
X.	RESULTADOS	18
XI.	DEFINICIONES Y ABREVIATURAS	20
XII.	REFERENCIAS	21

Manual de usuario

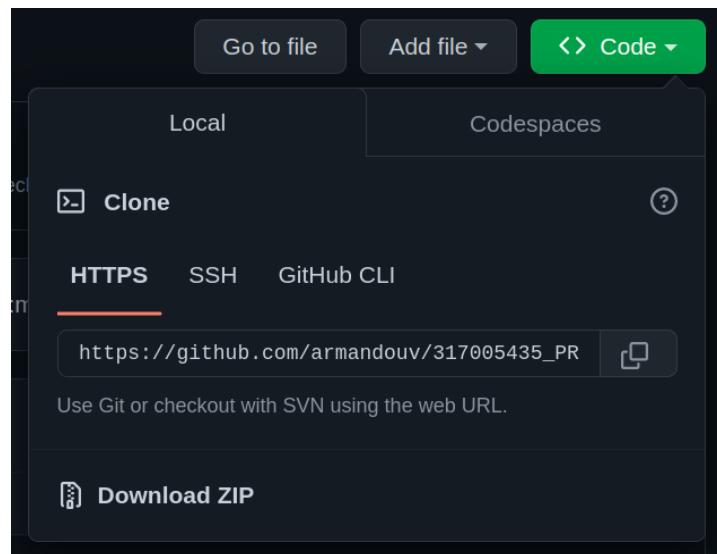
I. Introducción

En este manual de usuario se presenta un breve tutorial de la descarga, compilación y ejecución del proyecto, así como las instrucciones necesarias para interactuar con el entorno gráfico de la casa de CJ, del videojuego Grand Theft Auto: San Andreas. Lo anterior incluye la forma de manipular la cámara en el entorno, así como las teclas correspondientes para efectuar las interacciones implementadas.

II. Descarga, compilación y ejecución

Para descargar el proyecto, es necesario ir a la página web del repositorio (https://github.com/armandouv/317005435_PROYECTOFINAL2023-1_GPO6).

Posteriormente, se debe hacer click en el botón Code, y luego en la opción descargar ZIP.



Una vez descargado, descomprimimos el archivo, y observamos que tenemos todos los archivos presentes en el repositorio.

En este punto, tenemos dos opciones para ejecutar el proyecto:

1. Compilar y ejecutar el proyecto

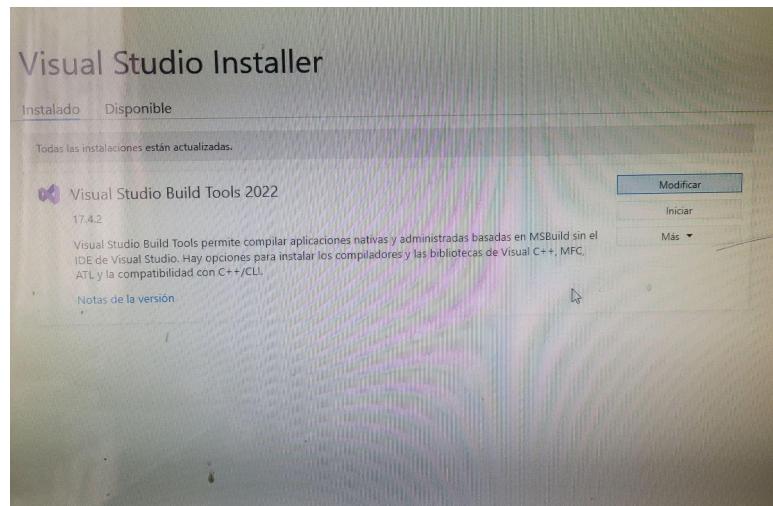
Si se desea compilar el proyecto, es necesario tener instaladas:

- Las herramientas para desarrollo en C++ de Visual Studio, que se pueden instalar de la siguiente forma:
 - Primero, hay que instalar Git, así como el instalador de las herramientas de desarrollo, utilizando los siguientes comandos:

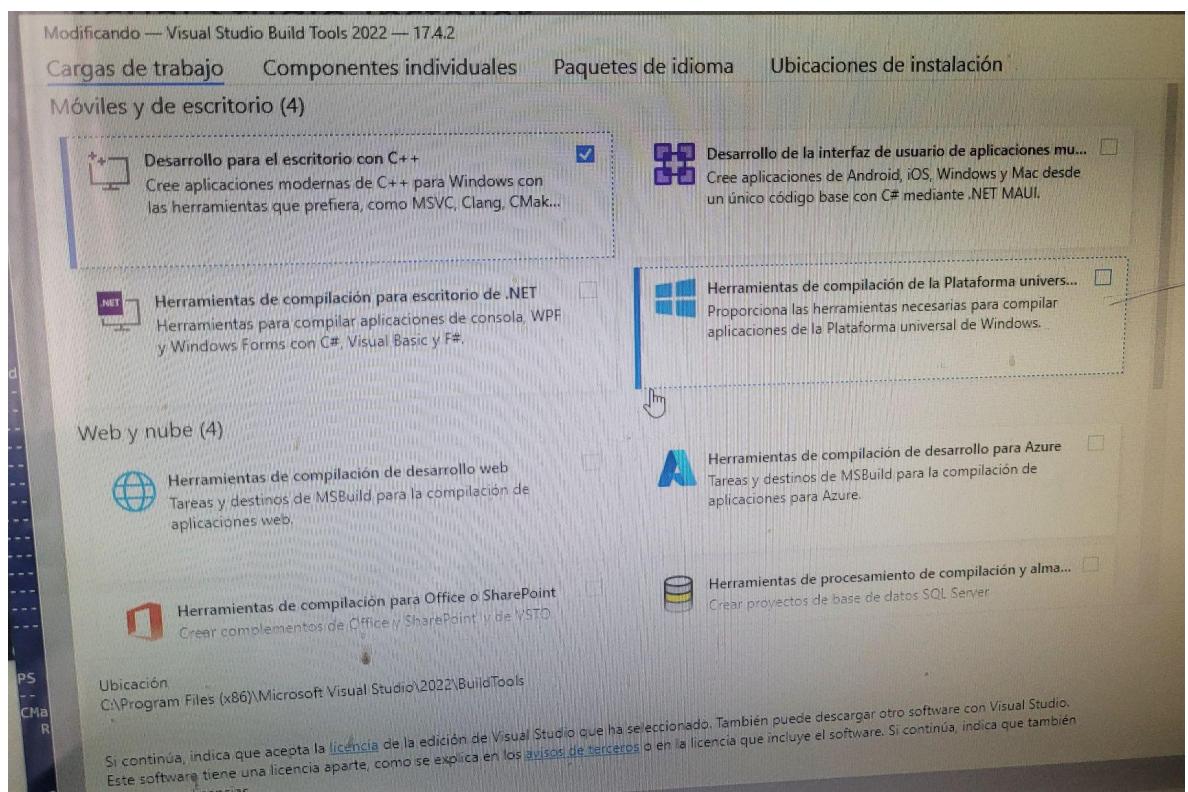
```
winget install -e --id Git.Git
```

```
winget install Microsoft.VisualStudio.2022.BuildTools
```

- Posteriormente, es necesario ejecutar el instalador de Visual Studio (o Visual Studio Installer), y seleccionar la opción Modificar.



- Finalmente, seleccionamos la opción de Desarrollo para el escritorio con C++, y hacemos click en Modificar. Esperamos que la instalación termine.



- La herramienta CMake, que, si no se instaló en el paso anterior, se puede instalar mediante el comando de Powershell:

```
winget install -e --id Kitware.CMake
```

Una vez que se tienen instaladas dichas herramientas, se deben ejecutar los siguientes comandos desde el **Powershell de desarrollo para Visual Studio**, recién instalado. Desde el directorio raíz del proyecto, se generará el ejecutable mediante los siguientes comandos. Es importante mencionar que este proceso puede tardar algunos minutos:

```
cd src  
cmake .  
cmake --build . --config Release -j4
```

Una vez hecho esto, podemos ejecutar el programa de la siguiente forma:

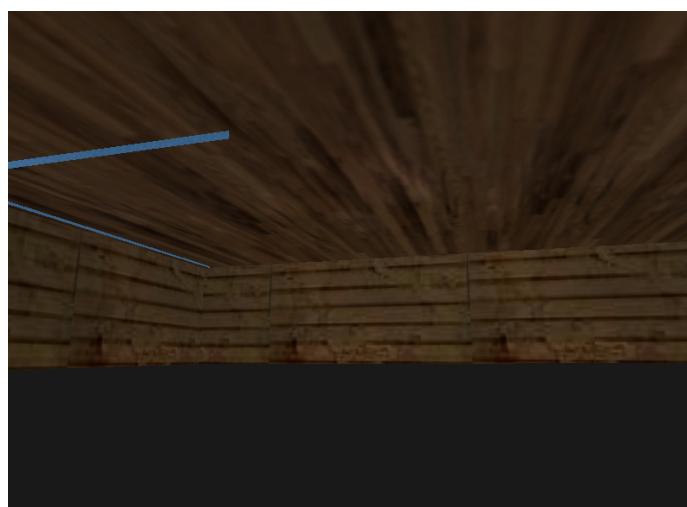
```
./proyecto_final.exe
```

2. Utilizando el ejecutable proporcionado

En el directorio raíz del proyecto, se encuentra un archivo con el nombre build.zip con el ejecutable del proyecto. Para hacerlo, es necesario primero descomprimir dicho archivo a un nuevo directorio build, y posteriormente ejecutar los siguientes comandos:

```
cd build  
./proyecto_final.exe
```

Una vez que se ejecute el programa, se mostrará una ventana con el entorno listo:



III. Interacciones

Se puede interactuar con el entorno virtual de la siguiente forma:

Cámara y controles generales

Mover adelante - Tecla de flecha arriba o W.

Mover atrás - Tecla de flecha abajo o S.

Mover a la izquierda - Tecla de flecha izquierda o A.

Mover a la derecha - Tecla de flecha derecha o D.

Rotación - Mouse

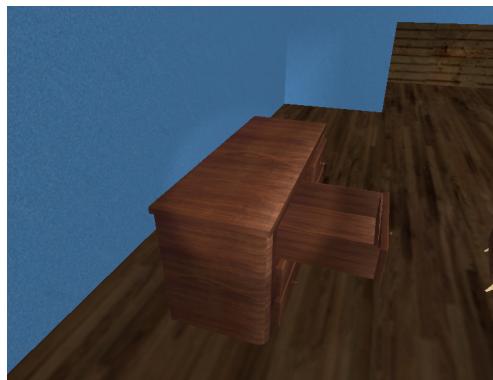
Salir - Tecla Esc

Entorno

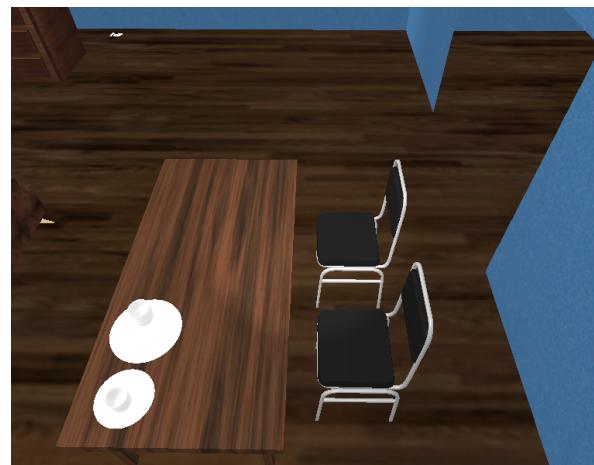
Activar y desactivar lámparas de techo - Tecla espacio.



Abrir y cerrar cajón - Tecla M.



Mover silla abajo y afuera de la mesa - Tecla N.



Abrir y cerrar puerta principal - Tecla B.



Manual técnico

I. Descripción del proyecto

Se deberá seleccionar una fachada y un espacio para su recreación 3D en OpenGL, que pueden ser reales o ficticios. Además, se deberán presentar imágenes de referencia de dichos espacios para su comparación y establecer. En la imagen de referencia se deberán visualizar 7 objetos que el alumno recreará virtualmente. Dichos objetos deberán ser lo más parecidos a la imagen de referencia, así como la ambientación incluida. Además, se deberán realizar 5 animaciones como mínimo: 3 sencillas y 2 complejas.

El proyecto se deberá entregar de forma individual, con un manual de usuario donde se explique cada interacción dentro del ambiente virtual recreado y un manual técnico que contenga la documentación del proyecto que incluya objetivos, diagrama de Gantt, alcance del proyecto, limitantes y la documentación del código. Por otro lado, se debe compartir la liga del proyecto en un repositorio en GitHub.

El espacio elegido para recrear es la casa de CJ, del videojuego Grand Theft Auto: San Andreas. Las imágenes de referencia propuestas son:

Fachada



Cuarto



Los elementos a representar, además de la ambientación son:

- Mesa de sala y mesa trasera
- Sillón pequeño (x2)
- Sillón grande
- Tocador
- Silla (x2)
- Trastes encima de mesa de la parte trasera (x2).
- Puerta
- Mosca
- Lámpara de techo
- Maceta con planta
- Araña

En la planeación inicial se consideraron únicamente 7 objetos, pero se aumentaron para mejorar la completitud del proyecto.

Las animaciones propuestas son:

- Abrir y cerrar puerta (simple)
- Abrir y cerrar cajón de tocador (simple)
- Mover silla debajo y fuera de la mesa (simple)
- Araña colgando de una telaraña (compleja)
- Mosca volando (compleja)

II. Objetivos y consideraciones

- El alumno deberá aplicar y demostrar los conocimientos adquiridos durante todo el curso, los cuales incluyen:
 - Modelado geométrico y jerárquico.
 - Conocimiento de un software de modelado de propósito específico como Maya.
 - Texturizado.

- Transformaciones básicas.
 - Animación sencilla y compleja.
 - Carga y adaptación de modelos
 - Iluminación y sombreado
- El entorno virtual recreado debe ser lo más parecido posible a la imagen de referencia presentada.
 - Se deben crear 5 animaciones como mínimo, teniendo 3 sencillas y 2 complejas. Además, cada animación debe tener contexto y coherencia.

III. Alcance

El proyecto apunta a ser una recreación del entorno virtual ya presentado, que sea fácil de utilizar y de instalar en cualquier computadora (véase el manual de usuario), por lo cual, se incluye:

- Ambiente virtual funcional de la fachada y cuarto seleccionados.
- Ambientación en el cuarto y fachada seleccionados.
- 12 objetos modelados.
- Interacciones con el entorno.
- Animaciones atractivas y coherentes.
- Manejo intuitivo de la cámara.
- Fluidez en la ejecución del programa.

Por otro lado, el proyecto no incluye:

- Recreación detallada del entorno exterior a la fachada.
- Modelos de objetos complejos (no considerados low poly).
- El uso de un Skybox.
- Animaciones utilizando técnicas avanzadas.

IV. Limitantes

- El entorno recreado se realizó utilizando una escala aproximada, ya que no se contó con un plano detallado con los tamaños apropiados, por lo que es probable que ésta no sea perfecta, lo cual influye en la distribución y tamaño de los objetos.
- Es posible que el usuario recorra partes del entorno donde no se encuentra ningún objeto de interés.
- No fue posible utilizar modelos muy complejos, y se optó por utilizar aquellos de poligonaje bajo para evitar problemas de rendimiento.
- Si bien las animaciones funcionan de forma correcta, es posible optimizar algunas de ellas para mejorar su rendimiento, ya que puede verse mermado al escalar el proyecto.
- Si la aplicación se despliega, el usuario final puede acceder a los Shaders empleados y no empleados, así como también los modelos ocupados; por lo que no existe un control sobre los archivos que puede y no ver un usuario final.

V. Diagrama de Gantt

+ Diagrama de Gantt

Tarea	Duración											
	Septiembre			Octubre			Noviembre					
	14	20 - 26	27	26	1	8	13	14	20	23	27	29
Presentación de proyecto y requerimientos	14	20 - 26	27	26	1	8	13	14	20	23	27	29
Selección de fachada y habitación		20 - 26										
Entrega de imagen de referencia			27				14	20	23	27	29	
Creación de modelo de mesa				26	1							
Creación de modelo de sillón grande					8							
Creación de repositorio de Github						13						
Agregar instrucciones de compilación							14	20	23	27	29	
Automatizar dependencias							20	23	27	29		
Arregla dependencia de GLEW								20	23	27	29	
Creación de modelo de tocador								23	27	29		
Arregla luz ambiental									27	29		
Creación de modelo de silla										27	29	
Creación de modelo de maceta										27	29	
Creación de modelo de trastes											27	29
Creación de fachada											27	29
Animación de cajón												29
Animación de silla												29

Tarea	Diciembre					
	1	2	4	5	6	7
Presentación de proyecto y requerimientos						
Selección de fachada y habitación						
Entrega de imagen de referencia						
Creación de modelo de mesa						
Creación de modelo de sillón grande						
Creación de repositorio de Github						
Agregar instrucciones de compilación						
Automatizar dependencias						
Arregla dependencia de GLEW						
Creación de modelo de tocador						
Arregla luz ambiental						
Creación de modelo de silla						
Creación de modelo de maceta						
Creación de modelo de trastes						
Creación de fachada						
Animación de cajón						
Animación de silla						
Creación de modelo de mosca						
Creación de modelo de araña						
Animación de mosca						
Animación de araña						
Creación de modelo de puerta						
Animación de puerta						
Creación de modelo de lámpara de techo						
Agrega point lights						
Crea ejecutable						
Elimina movimiento de point lights						
Elaboración de documentación						
Entrega de proyecto						

VI. Requerimientos

Los requerimientos formales del proyecto son los siguientes:

- Tener una imagen de referencia de la fachada y cuarto, y especificar los 7 objetos a modelar.
- Realizar una recreación 3D en OpenGL de un espacio que puede ser real o ficticio. Se eligió la casa de CJ, del videojuego Grand Theft Auto: San Andreas.
- Se deberá incluir cámara sintética.
- Se deberá realizar un manual de usuario y un manual técnico, en inglés y español.
- Deberán implementarse por lo menos 3 animaciones sencillas y dos complejas, que deben de tener contexto y ser coherentes.
- Los objetos y ambientación deberán ser lo más parecidas posibles a la imagen de referencia.
- Se deberá colocar el proyecto en un repositorio en Github.

- Deberá entregarse un archivo ejecutable.
- Debe hacerse un análisis de costos.

VII. Diseño del sistema

Desde el directorio src del proyecto, se pueden diferenciar los distintos módulos, tanto en los distintos archivos de código fuente como en los subdirectorios correspondientes. Los módulos principales que componen al sistema son los siguientes:

- **Módulo de Modelos:** En él se encuentran los modelos 3D utilizados en el proyecto, y se ubican en el directorio Models. Cabe mencionar que cada modelo se encuentra en formato obj, con sus respectivos archivos de texturas y mtl. Posteriormente, en el módulo principal se realizará la carga de modelos para cada uno cuando se requiera.
- **Módulo de Shaders:** En él se encuentran los Shaders que se utilizan en el proyecto, además de algunos extras para futura expansión del proyecto. No todos se están implementando, sin embargo, se deciden conservar para futuras referencias. El shader principal utilizado en el proyecto es el lightingShader, que calcula la iluminación adecuada para la recreación del espacio.
- **Módulo de archivos auxiliares:** Aquí se consideran todos los archivos con extensión .h presentes en el proyecto, cuyos objetivos son implementar algunas funcionalidades auxiliares necesarias, como la implementación del manejo de la cámara, carga de modelos, shaders, texturas, etc.
- **Módulo principal (main):** En el módulo principal, se implementa toda la lógica principal del proyecto, lo cual incluye la configuración de OpenGL para el renderizado, carga e inicialización de las librerías necesarias, creación de ventanas, carga de shaders, modelos y texturas. Además, se implementan las animaciones realizadas, se realizan transformaciones básicas, se configura la iluminación del entorno, se dibujan los modelos importados utilizando el lightingShader, y se implementan las interacciones con el entorno especificando las teclas adecuadas o entrada con el mouse para definir el comportamiento.

VIII. Especificaciones funcionales

Como se mencionó, en el módulo principal se implementa toda la lógica principal del proyecto, lo cual incluye:

- Configuración de OpenGL para el renderizado
- Carga e inicialización de las librerías necesarias

- Creación de ventanas
- Carga de shaders, modelos y texturas
- Transformaciones básicas para adaptación de modelos
- Implementación de iluminación del entorno
- Dibujo de modelos importados utilizando el lightingShader
- Implementación de animaciones
- Implementación de las interacciones con el entorno especificando las teclas adecuadas o entrada con el mouse para definir el comportamiento.

A continuación, se muestra el desglose de cada una de las funciones mencionadas, dentro de funciones específicas en el código:

Nombre de función	Especificación
main()	<p>Se trata de la función principal dentro del código, como su nombre lo indica.</p> <p>Primero, se inicia configurando la ventana a utilizar, con ayuda de las utilidades que provee la biblioteca GLFW, y se inicializa la biblioteca GLEW. Posteriormente, se crean e inicializan los Shaders a utilizar, así como los modelos, con ayuda del módulo de archivos auxiliares. Se configuran los VAO y VBO con ayuda de OpenGL, y, en el loop de renderizado se ejecuta la mayor parte de tareas presentadas anteriormente. Éstas incluyen la implementación de la lógica necesaria para las animaciones, configuración de iluminación incluyendo luz direccional y point lights, y adaptación y dibujo de modelos utilizando el lightingShader y transformaciones básicas.</p>
DoMovement()	<p>En esta función se lidia con el movimiento de la cámara al presionar las teclas adecuadas, utilizando el módulo de Camera respectivo.</p>
KeyCallback(GLFWwindow* window, int key, int scancode, int action, int mode)	<p>En esta función se implementan varios aspectos de la interacción con el entorno: el cerrado de la ventana y terminación del programa, al presionar la tecla ESC, el prendido y apagado de las lámparas de techo, utilizando la tecla espacio, y la activación de las animaciones del</p>

	cajón, silla y puerta, con las teclas m, n y b, respectivamente.
MouseCallback(GLFWwindow* window, double xPos, double yPos)	En esta función se lida con la rotación de la cámara al presionar las teclas adecuadas, utilizando el módulo de Camera respectivo.

IX. Análisis de costos

Para determinar el costo final del proyecto, se toman en cuenta todas las actividades realizadas, y se realiza un análisis objetivo, tomando en cuenta tanto el valor del producto generado como del tiempo requerido para realizarlo. Se considera un costo de trabajo por hora de 90 pesos mexicanos. Para determinar el precio final de cada rubro, se considera el costo de trabajo por hora y se realiza un análisis de precios de mercado de ser necesario (por ejemplo, para la realización de los modelos, se comparan precios con productos de la plataforma Turbosquid):

- *Propuesta de imagen de referencia de la fachada y cuarto, y 7 objetos a modelar.*
Tiempo estimado: 4 horas.
Costo estimado: 360 pesos mexicanos.
- *Implementación y adecuación de modelos (se toma en cuenta el tiempo de trabajo, así como los precios de mercado para cada modelo)*
 - *Adecuación*
Tiempo estimado: 6 horas
Costo estimado: 540 pesos mexicanos.
 - *Mesa de sala y mesa trasera*
Tiempo estimado: 1 hora
Costo estimado: 100 pesos mexicanos.
 - *Sillón pequeño (x2)*
Tiempo estimado: 1.5 horas
Costo estimado: 180 pesos mexicanos.
 - *Sillón grande*
Tiempo estimado: 2 horas
Costo estimado: 200 pesos mexicanos.

- *Tocador*
Tiempo estimado: 2 horas
Costo estimado: 230 pesos mexicanos.
- *Silla (x2)*
Tiempo estimado: 1.5 horas
Costo estimado: 150 pesos mexicanos.
- *Trastes encima de mesa de la parte trasera (x2).*
Tiempo estimado: 1 hora
Costo estimado: 100 pesos mexicanos.
- *Puerta*
Tiempo estimado: 1 hora
Costo estimado: 110 pesos mexicanos.
- *Mosca*
Tiempo estimado: 2 horas
Costo estimado: 300 pesos mexicanos.
- *Lámpara de techo*
Tiempo estimado: 1.5 horas
Costo estimado: 160 pesos mexicanos.
- *Maceta con planta*
Tiempo estimado: 3 horas
Costo estimado: 400 pesos mexicanos.
- *Araña*
Tiempo estimado: 3 horas
Costo estimado: 450 pesos mexicanos.

Costo estimado: 2920 pesos mexicanos

- *Implementación de animaciones*
Tiempo estimado: 5 horas.
Costo estimado: 450 pesos mexicanos.
- *Realización de manual técnico y de usuario*
Tiempo estimado: 7 horas.
Costo estimado: 630 pesos mexicanos.

Costo total estimado del proyecto: 4360 pesos mexicanos

X. Resultados

A continuación, se muestra una breve comparación de las imágenes de referencia propuestas y el entorno final obtenido, como muestra del resultado del proyecto:

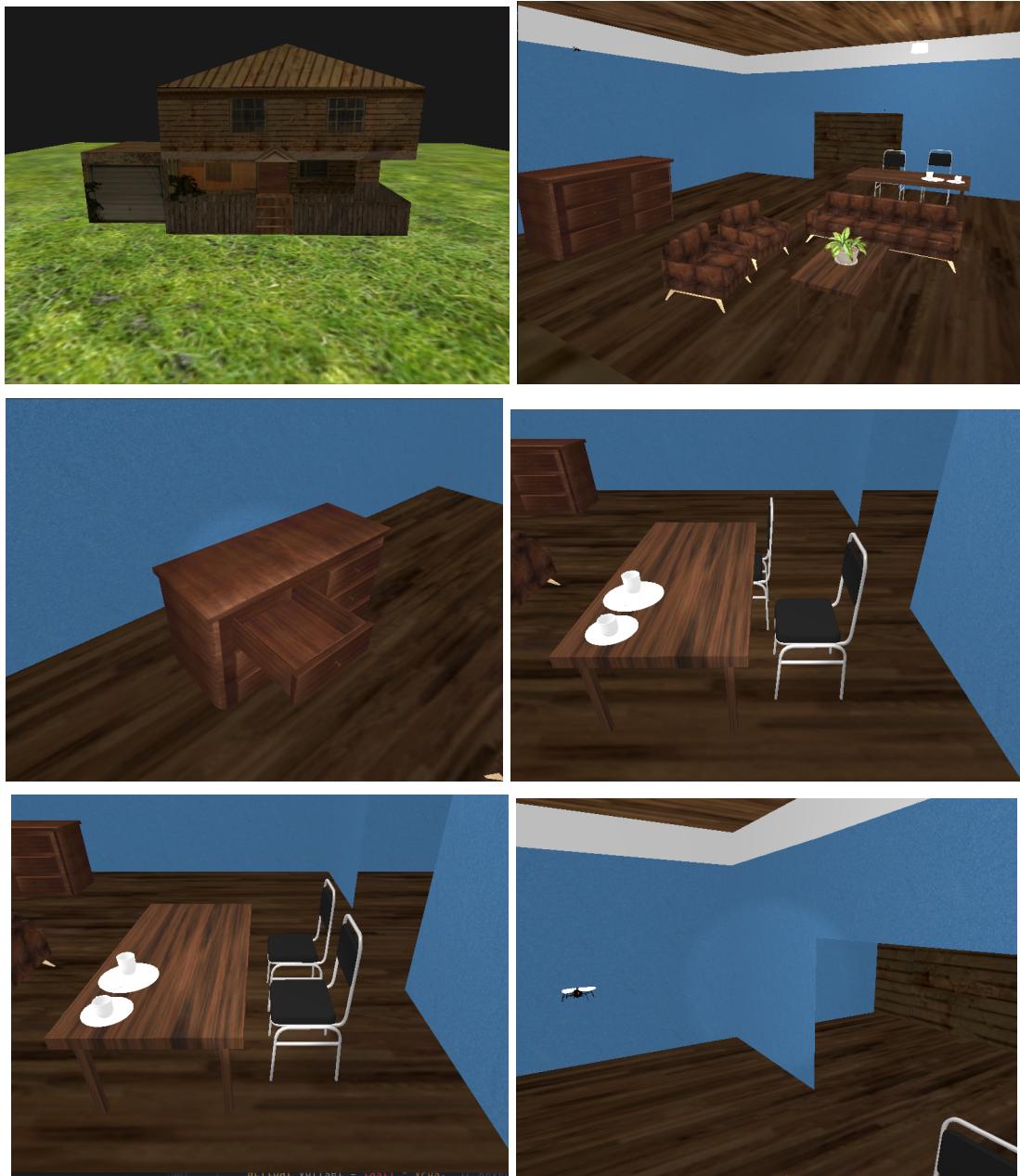
Fachada

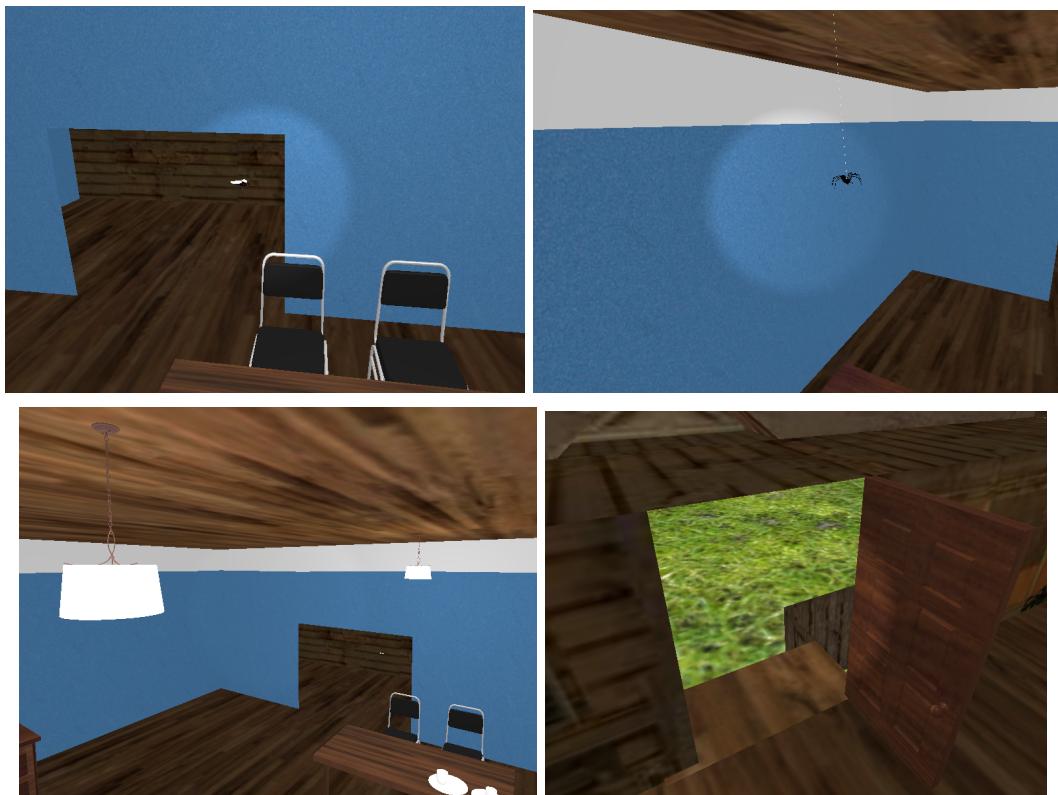


Cuarto



Recreación





XI. Definiciones y abreviaturas

Skybox	Es un elemento gráfico que permite dar, en un espacio tridimensional, la ilusión de que este espacio es más extenso de lo que realmente es, al proporcionar un efecto visual de un escenario por medio de una textura aplicada a un objeto dentro del cual se encuentra ubicada la cámara del entorno.
Textura	Es una imagen mapeada en el espacio UV que es asociada a una malla tridimensional para darle mayor realismo.
Archivo .obj	Es un formato de definición de geometría. Representa sólo la geometría 3D, es decir, la posición de cada vértice, la posición UV de cada vértice de coordenadas de textura, las normales de vértice y las caras que hacen que cada polígono se defina como una lista de vértices.
Archivo .mtl	Un archivo mtl es un archivo auxiliar para archivos obj, que contiene definiciones de materiales que podrían ser utilizados por el archivo obj. Cada definición de material empieza con una sentencia newmtl, que define el nombre del material, seguido de líneas especificando propiedades particulares.
Shader	Es un programa que calcula los niveles apropiados de luz, oscuridad y color durante el renderizado de una escena 3D. La mayoría corren en la GPU, pero no es un requerimiento estricto.
OpenGL	OpenGL es una biblioteca que proporciona servicios para renderizar gráficos 2D y 3D.

GLFW	<i>OpenGL FrameWork o Graphics Library FrameWork</i> , es una biblioteca ligera de utilidades para usar con OpenGL. Sus siglas significan Graphics Library Framework. Brinda a los programadores la capacidad de crear y administrar ventanas y contextos OpenGL, así como manejar la entrada de joystick, teclado y mouse.
GLEW	Es una biblioteca de código abierto para C/C++, cuya funcionalidad es la carga de extensiones de OpenGL. Proporciona mecanismos de tiempo de ejecución eficientes para determinar qué extensiones de OpenGL son compatibles con la plataforma de destino. La funcionalidad principal y de extensión de OpenGL se exponen en un solo archivo de encabezado.
GLSL	<i>OpenGL Shading Language</i> , es el principal lenguaje de shading para OpenGL. Se encuentra basado en C.
GLM	<i>OpenGL Mathematics</i> , Es una biblioteca matemática escrita en C++ para el desarrollo de software gráfico basado en OpenGL. Permite manipular fácilmente vectores y matrices y realizar operaciones con ellas.

XII. Referencias

- Nash Vail. (2017). *Understanding Linear Interpolation in UI Animation*. 28 de agosto de 2022, de FreeCodeCamp. Sitio web: <https://www.freecodecamp.org/news/understanding-linear-interpolation-in-ui-animations-74701eb9957c/>
- Foley, J. D, Van Dam, A. (1982). *Fundamentals of Interactive Computer Graphics*. 28 de agosto de 2022, de Addison-Wesley. Sitio web: <https://archive.org/details/fundamentalsofin00fol>
- Desconocido. (2014). *Uniform Variables*. 28 de agosto de 2022, de Cornell University. Sitio web: <https://www.cs.cornell.edu/courses/cs4620/2014fa/lectures/glsl2.pdf>
- Varios autores. (2022). *Frequently Asked Questions*. 28 de agosto de 2022, de GLFW Sitio web: <https://www.glfw.org/faq.html>
- Varios autores. (2017). *The OpenGL Extension Wrangler Library*. 28 de agosto de 2022, de GLEW Sitio web: <http://glew.sourceforge.net/>
- Orlando Saldívar Esquivel. (2021). *Plantilla de la materia de Administración de Proyectos*. Consultado el 2 de diciembre de 2022.
- Varios autores. (2018). *MTL Files, Material Definitions for OBJ Files*. Consultado el 26 de septiembre de 2022, de Florida State University. Sitio web: <https://people.sc.fsu.edu/~jburkardt/data/mtl/mtl.html>
- Mike Bailey. (2022). *OBJ Files*. Consultado el 26 de septiembre de 2022, de Oregon State University. Sitio web: <https://web.engr.oregonstate.edu/~mjb/cs550/PDFs/ObjFiles.4pp.pdf>
- Andy Johnson. (2022). *Texture and Other Mapping*. Consultado el 26 de septiembre de 2022, de University of Illinois Chicago. Sitio web: <https://www.cs.uic.edu/~jbell/CourseNotes/ComputerGraphics/TextureMapping.html>