



Universidad nacional autónoma de México.

Facultad de ingeniería.

Proyecto Final.

Materia: Computación grafica e interacción humano-computadora –
Teoría.

Profesor: Ing. Carlos Aldair Roman Balbuena.

Alumno:

Maceda Patricio Fernando
Núm. Cta. 316054416

Semestre: 2022-2.

Fecha de entrega: 27 de mayo de 2022.

Proyecto final de teoría.

Contenido.

Objetivo.	3
Requerimientos.	3
Introducción.	4
Desarrollo.	6
Información de los modelos.	6
Descargadas de internet:	6
Realizadas por el equipo:	12
Cronograma de actividades.	20
Información de bibliotecas, shaders y funciones.	21
Información de las animaciones.	24
Estudio técnico para la determinación del costo del proyecto:	32
Costo directo:	34
Costo indirecto:	34
Costo total:	35
Manual de usuario y Mapa de navegación:	35
Video explicativo:	35
Limitantes.	35
Conclusiones.	36
Referencias.	36

Objetivo.

El estudiante realizará el proyecto final de teoría de Computación gráfica e interacción humano-computadora, en el cual aplicará los conceptos vistos a lo largo del curso. Por el proyecto final de laboratorio ya se tiene modelo un conjunto habitacional real junto con 5 animaciones el cual se realizó en equipo de 4 personas, para cumplir los requerimientos del proyecto de teoría se agregarán dos cuartos amueblados con animaciones extra.

Requerimientos.

- El alumno deberá de realizar un ambiente virtual, el cual representará dos cuartos, formado por cinco elementos a recrear los cuales ya fueron aprobados por el profesor.
- Integrar cámara sintética.
- Debe contener por lo menos 4 animaciones
- Se deberá entregar la documentación del proyecto tanto en español como en inglés (diagrama de Gantt, manual de usuario donde se plantean objetivos e interacción del proyecto y manual técnico)
- Se debe agregar un análisis de costos del proyecto.
- La entrega de documentación se debe hacer en formato digital.
- Debe entregarse un archivo ejecutable.
- Se evaluará el realismo del espacio.
- Debe estar el proyecto en GitHub.

Introducción.

Para este proyecto se escogió modelar la unidad habitacional Villa Olímpica, ubicada en la alcaldía Tlalpan de la Ciudad de México, en específico en el predio de nueve hectáreas entre la Avenida Insurgentes Sur y la Avenida Periférico, muy cerca de Ciudad Universitaria.

El complejo fue construido por el Banco Nacional de Obras y Servicios Públicos para alojar a los Atletas de la Olimpiada de México, consta de veintinueve edificios que reúnen un total de 5,044 habitaciones y 2,572 baños en 904 departamentos, del total de los inmuebles, de ellos 24 se destinaron para los competidores varones y 3 para las mujeres atletas, los restantes dos fueron empleados para la prensa (Edificios de México, (s. f.)).

Se escogió este lugar en principio porque un compañero del equipo vive ahí, por lo cual se facilitaba el modelado de los objetos que se colocaran en el proyecto. Aprovechando que los edificios tienen la misma forma, sólo habría falta el diseño de un edificio y duplicarlo las veces que se desee, además, este lugar nos permite la colocación de diversos elementos que nos permite cubrir los requerimientos impuestos por el profesor. Cabe resaltar que nos dimos la libertad de modificar algunos objetos de manera que pudiera concordar con nuestra visión, cómo reducir el número de edificios, la colocación de una tienda OXXO y una cancha genérica.

Los cuartos para modelar se obtuvieron de internet, son cuartos con temática de estudiante los cuales cuentan con una cama y unos pocos muebles, se adjuntan las imágenes de referencia en las que me base para modelar los cuartos y los elementos.

Finalmente mostraremos el diseño con la disposición de los elementos que entregamos en la propuesta del proyecto.

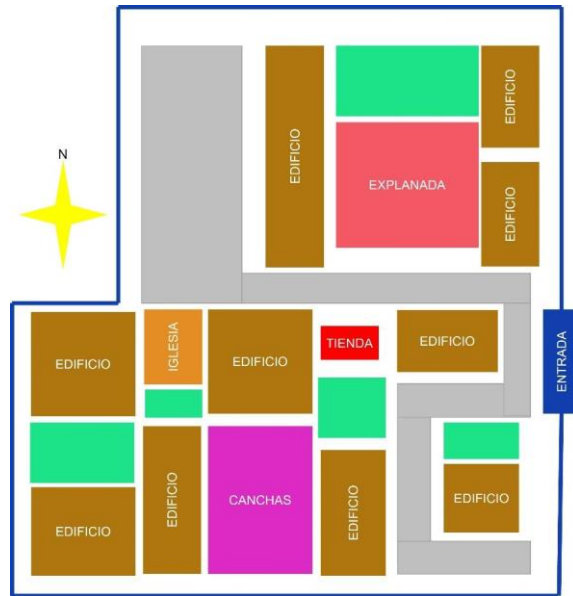


Figura 1. Mapa de la propuesta.



Figura 2. Cuarto 1 propuesto.



Figura 3. Cuarto 2 propuesto.

Desarrollo.

Información de los modelos.

A continuación, mostraremos los datos de descarga de los modelos obtenidos de internet y algunos detalles de los modelos realizados por nosotros.

Descargadas de internet:

Modelo: Electric Floor Fan 3D Model.

Animación. Animación de aspas girando.

Sitio de internet: open3dmodel.com.



Figura 4. Ventilador.

Modelo: Hombre en chaqueta de pie y Rigged 3D Modelo.

Animación. Persona que conduce el triciclo.

Sitio de internet: open3dmodel.com



Figura 5. Hombre de pie.

Modelo: Man 3D Model.

Animación. Persona que conduce una bicicleta.

Sitio de internet: archive3d.net



Figura 6. Hombre en bicicleta.

Modelo: Mini fridge.

Animación. Animación de puerta abriéndose.

Sitio de internet: 3dsky.org



Figura 7. Refrigerador.

Modelo: Pete

Animación. Hombre que camina hacia el oxo (Left Strafe Walking).

Sitio de internet: Mixamo.



Figura 8. Hombre caminando de lado.

Personaje: Samoyedo Perro Modelo 3D

Animación. Perro que camina por la unidad.

Sitio de internet: open3dmodel.com



Figura 9. Perro.

Personaje: Shannon.

Animación: Persona corriendo al estilo del Fútbol.

Sitio de internet: [Mixamo.](http://Mixamo.com)



Figura 10. Deportista.

Modelo: Sporty Granny

Animación. Persona caminando por la unidad (Female Walk)

Sitio de internet: Mixamo.



Figura 11. Mujer deportista.

Modelo: Triciclo de reparto 2.

Animación. Un triciclo que simula un puesto de tamales.

Sitio de internet: 3dwarehouse.sketchup.com.

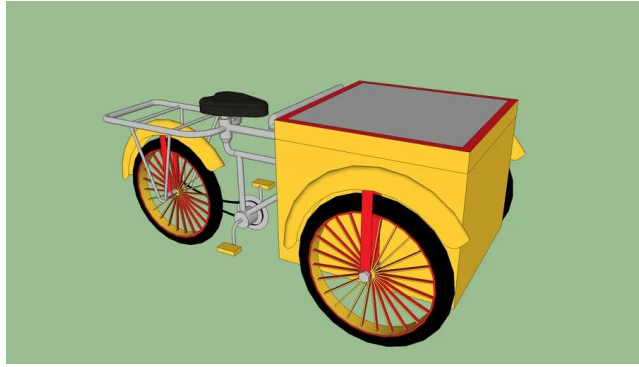


Figura 12. Triciclo.



Figura 13. Imagen de un mundo de tamal.

Modelo: Vintage Vw Volkswagen Beetle Car Modelo 3D.

Animación. Carro que simula conducirse por la unidad.

Sitio de internet: Open3DModel.



Figura 14. Bocho.

Realizadas por el equipo:

Modelo: Buro.

Aplicación de diseño: Autodesk 3ds Max.



Figura 15. Buro.

Modelo: Cama 1.

Aplicación de diseño: Autodesk 3ds Max.



Figura 16. Cama 1.

Modelo: Cama 2.

Aplicación de diseño: Autodesk 3ds Max.



Figura 17. Cama 2.

Modelo: Cancha.

Aplicación de diseño: Blender.

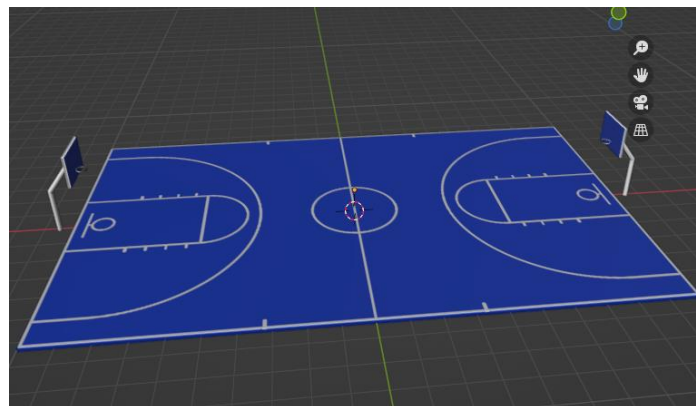


Figura 18. Cancha.

Modelo: Cuadro 1.

Aplicación de diseño: Autodesk 3ds Max.



Figura 19. Cuadro 1.

Modelo: Cuadro 2.

Aplicación de diseño: Autodesk 3ds Max.

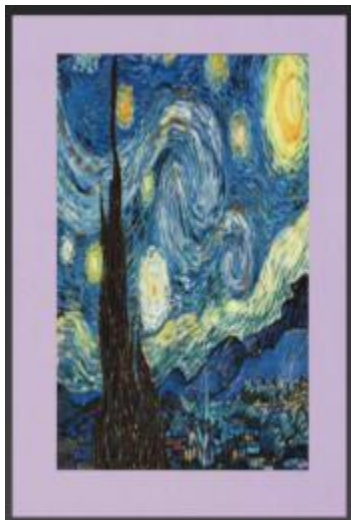


Figura 20. Cuadro 2.

Modelo: Edificio.

Aplicación de diseño: Blender.

El modelo corresponde a los edificios ubicados en la unidad. Para modelarlo se utilizaron planos y cubos. Con sencillas transformaciones y extrusiones se pudo lograr el acabado similar a nuestras referencias originales.

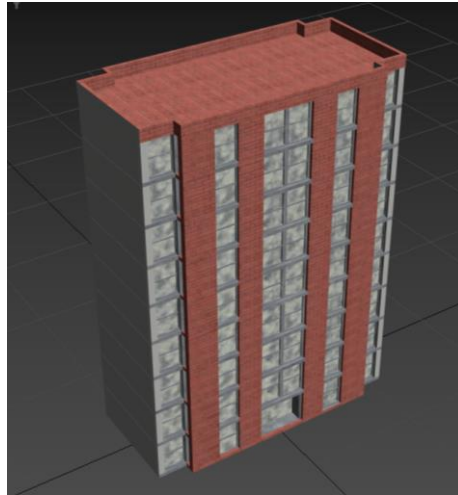


Figura 21. Edificio.

Modelo: Entrada.

Aplicación de diseño: Autodesk 3ds Max.

Este modelo corresponde a la entrada a la unidad. Solo se utilizaron figuras primitivas y un “TexPlus” para los números que indican la dirección.



Figura 22. Entrada.



Figura 23. Fachada real de la entrada.

Como se ve en la imagen de referencia colocada en la propuesta, aunque para el modelo se ocupó una textura de un color más oscuro.

Modelo: Librero.

Aplicación de diseño: Autodesk 3ds Max.



Figura 24. Librero.

Modelo: Mesa.

Aplicación de diseño: Autodesk 3ds Max.



Figura 25. Mesa.

Modelo: Iglesia.

Aplicación de diseño: Blender.

El modelo de la iglesia se modeló usando también figuras simples como cubos a los cuales se les aplicaron transformaciones de escala y traslación. Para el vitral únicamente se colocó una textura sacada de una fotografía tomada a la iglesia física.

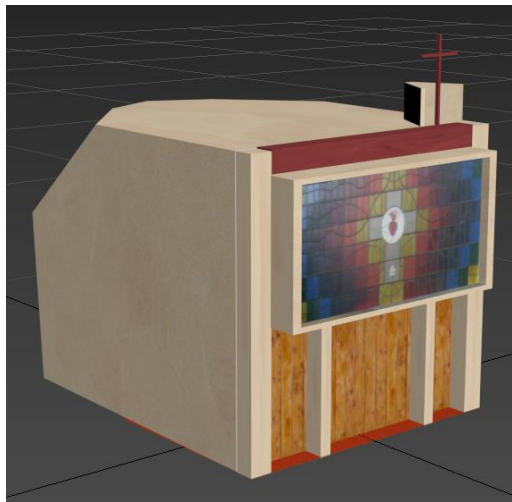


Figura 26. Iglesia.

Modelo: OXXO.

Aplicación de diseño: Autodesk 3ds Max.

El modelo corresponde a una sucursal de las tiendas Oxxo dentro de la unidad. Para el modelado se utilizaron planos, cajas y cilindros además de utilizar sus colores característicos para la textura.



Figura 27. OXXO.

Para la fachada se utilizó la imagen “Fachada de Tienda OXXO” descargada de la página “flickr.com” y subida por la cuenta “FEMSA_Corporativo”.



Figura 28. Fachada Oxxo.

Modelo: Silla.

Aplicación de diseño: Autodesk 3ds Max.



Figura 29. Silla.

Modelo: Televisión.

Aplicación de diseño: Autodesk 3ds Max.



Figura 30. Televisión.

Proyecto Final Teoria

Inicio del proyecto:

vie, 4/1/2022

Semana para mostrar:

20

Tablero kanban.

The screenshot shows a GitHub Projects board for the repository 'Fernando1612 / Proyecto-Final-CGEIH-C'. The board is titled 'Proyecto final' and has a green progress bar indicating it is 'Updated now'. The board is divided into three columns: 'To do' (0 items), 'In progress' (0 items), and 'Done' (9 items). The 'Done' column contains two sections of tasks:

- Modelos y Animaciones Alexis** (Added by Fernando1612):
 - Cancha
 - Persona paseando perro (persona y perro) (C)
 - Persona caminando (E)
- Modelos y Animaciones Daniel** (Added by Fernando1612):
 - Edificio (remodelarlo y colocar 10)
 - Explanada, zonas verdes y caminos
 - Iglesia

Información de bibliotecas, shaders y funciones.

Biblioteca	Descripción
GLEW, GLFW	Para la comunicación con el SO para el manejo de ventanas en Visual Studio y OpenGL (creación, eliminación, etc.).
GLM	Biblioteca matemática de C++ para proyectos sobre la API OpenGL.
Assimp	Biblioteca para importar modelos 3d de programas como 3ds MAX, Maya, Blender, etc. y que OpenGL pueda entender su formato, en nuestro caso permite

Shader_m.h	Crea el vertex shader y fragment shader, lo único que recibe son dos direcciones, la primera para el archivo de vertex shader y la segunda con el archivo del fragment shader. Abre dichos archivos recuperando el código que contienen y posteriormente crea ambos shader asignándoles localidades de memoria, se compilan, verifica que se hayan creado correctamente y se crea el programa enlazando los shaders creados (VAO, VBO y EBO), compila, verifica la correcta creación y al final borra la información.
Camera.h	Crea la cámara sintética o interactiva, este .h define los movimientos que tendrá la cámara (adelante, atrás, izquierda y derecha), las posiciones o valores iniciales de la cámara para la rotación, inclinación, velocidad de movimiento, sensibilidad y zoom o ángulo de apertura para la proyección en perspectiva. Para los movimientos define listeners o callbacks los cuales van a actualizar los valores antes mencionados cuando se presionen las teclas indicadas o haya interacción con el mouse para el control de movimiento
Model.h	Recibe la dirección del archivo .obj y se encarga de importar dicho modelo utilizando assimp, cargando coordenadas de sus vértices (posiciones, normales y de texturas)
modelAnim.h	Recibe la dirección del archivo .dae y se encarga de importar dicho modelo utilizando assimp, cargando coordenadas de sus vértices (posiciones, normales y de texturas). Sirve para importar animaciones realizadas en otro software.

Shaders	Descripción
Skybox.frag y Skybox.vs	El vertex shader (.vs) captura la posición de la imagen y se la manda al fragment (.frag) para enlazarla al cubo.

lighting.vs y lighting.frag	El vertex shader accede a todas las coordenadas del VAO (posición, color y textura), recibe las matrices de modelo, vista y proyección para modificarlas según la geometría recibida del VAO. Se calcula el fragPos y las normales para pasarlo al fragment junto con las coordenadas de textura. El fragment shader define los tipos de luz que estarán disponibles (directional, point y spot) en forma de estructura con sus componentes. Define sus funciones para cada tipo de luz donde cada una calcula la iluminación en base a sus componentes difusa, especular, etc. con la información del vertex shader y las matrices del programa.
anim.vs y anim.frag	

Función	Descripción
my_input	Procesa la entrada por teclado, las teclas presionadas para realizar una acción. Se consideran 7 teclas en específico para la interacción con el ambiente virtual ("Esc", "w", "a", "s", "d", "z" y "space").
mouse_callback	Es el listener de la entrada por mouse. La acción que define para dicha entrada es la actualización en la rotación de la cámara
scroll_callback	Es el listener de la entrada por scroll del mouse. La acción que define para dicha entrada es hacer zoom a la cámara
animate	Esta función define las acciones a realizar si hay animaciones activadas iniciadas en KeyCallback, cuando cambian de estado y también la actualización de la cámara.
music	Función que sirve para activar el sonido con la tecla "z"

main	<p>Función principal con el programa para establecer el ambiente virtual. Crea la ventana, verifica errores, llama a las funciones “listener” o callbacks para los eventos de teclado y mouse. Define el viewport, activa las opciones de OpenGL para el manejo de materiales translúcidos y/o transparentes. Crea y configura los shaders. Define la geometría para las point light, skybox y plano de animación. Define las matrices de proyección, modelo, view. Captura los eventos de los listeners de movimiento y de entradas de teclado y mouse. Define la iluminación. Sucede la carga de modelos y transformaciones básicas para acomodarlos en el espacio o animarlos. Dibuja las point light. Manda a actualizar la cámara y mediante la captura de eventos está a la escucha para cuando se mande a terminar el programa.</p>
------	--

Información de las animaciones.

Animación de hombre en bici:

La animación está dividida en varios objetos:

- Hombre pedaleando
- Cuadro de la bicicleta
- Pedales
- Ruedas

La animación del hombre pedaleando fue creado con 3ds Max utilizando la herramienta de Auto key. al momento de dibujar al hombre en el programa le asignamos las variables de movimiento en x y z de la bicicleta y de rotación para que al momento que la bicicleta se mueva, el hombre también lo haga.

Dibujamos la bicicleta con sus valores iniciales y creamos un modelo temporal. Para la animación de los pedales y las ruedas agregamos una rotación del eje x:

Para la animación de la bici se asignó una ruta dentro del mapa en donde se utilizarán 6 estados del 0 al 5 utilizando la estructura de control Switch/Case, la

bicicleta va a cambiar su posición en el eje x y el eje z. Esta animación se repite infinitamente ya que al llegar al estado 5 la bicicleta se encuentra en las coordenadas de origen y regresa al estado 0 tampoco se necesita presionar alguna tecla para que comience, la animación inicia junto con el programa.

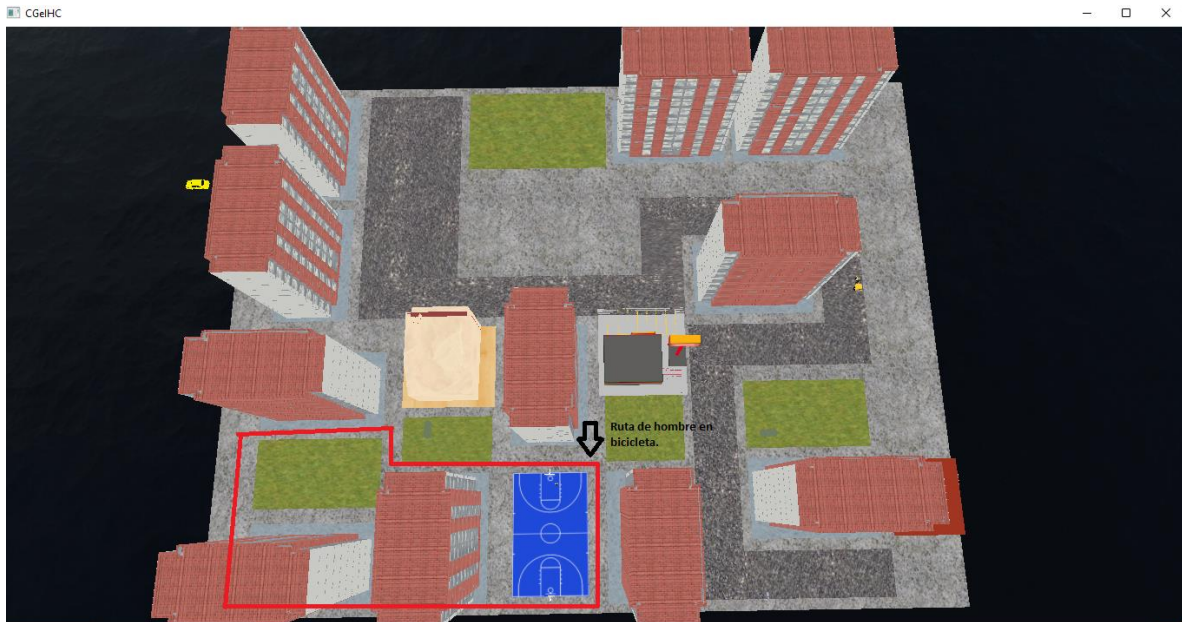


Figura 16. Ruta de animación de bicicleta.

Animación de hombre vendiendo tamales:

La animación está dividida en varios objetos:

- Hombre pedaleando
- Cuadro del triciclo
- Pedales
- Ruedas

La animación del hombre pedaleando fue creado con 3ds Max utilizando la herramienta de Auto key. al momento de dibujar al hombre en el programa le asignamos las variables de movimiento en x y z del triciclo y de rotación para que al momento que la bicicleta se mueva, el hombre también lo haga. Al momento de cargar el modelo tenía una rotación incorrecta por lo que se hicieron dos rotaciones, una al inicio para acomodarlo en el triciclo y la segunda que va a ir cambiando conforme el triciclo avance.

Dibujamos el triciclo con sus valores iniciales y creamos un modelo temporal. Para la animación de los pedales y las ruedas agregamos una rotación del eje x.

Para la animación del triciclo se asignó una ruta dentro del mapa en donde se recorre todo el conjunto habitacional utilizando 16 estados del 0 al 15, utilizando la estructura de control Switch/Case para los cambios en el eje x y z. Esta animación se repite infinitamente ya que al llegar al estado 15 el triciclo se encuentra en las coordenadas de origen y se regresa al estado 0, tampoco se necesita presionar alguna tecla para que comience, la animación inicia junto con el programa.



Figura 17. Ruta del señor de los tamales.

Animación de hombre haciendo deporte en la cancha:

La animación está formada por un único objeto, el personaje Shannon con la animación Running, descargada de la página Mixamo. Al momento de descargar se indicó que la animación debía ser en un solo lugar, para que a base de código se generará el desplazamiento. Para poder dibujar el objeto se debió colocar la variable que le permitirá desplazarse en el eje de las Z, esto tomando ventaja de que no se moverá en los ejes X y Y; en conjunto con una variable que permita su rotación sobre el eje Y para que simule su cambio de dirección.

Ocupando un SWITCH se escoge entre los diferentes estados, que representan hasta qué distancia de la cancha debe llegar el objeto, dentro de cada CASE se encuentra

una sentencia IF que verifica si el objeto se encuentra avanzando (hacia valores mayores) o regresando (hacia valores menores). En caso de que se cumpla la condición de que llegue hasta cierta distancia, el objeto realizará un giro de 180° y realizará el movimiento siguiente cambiando de estado.

Para que resulte la animación de que el personaje corra hasta cada marca y regrese hasta el origen una y otra vez, ya que no se condiciono la animación a una entrada del usuario al presionar una tecla.

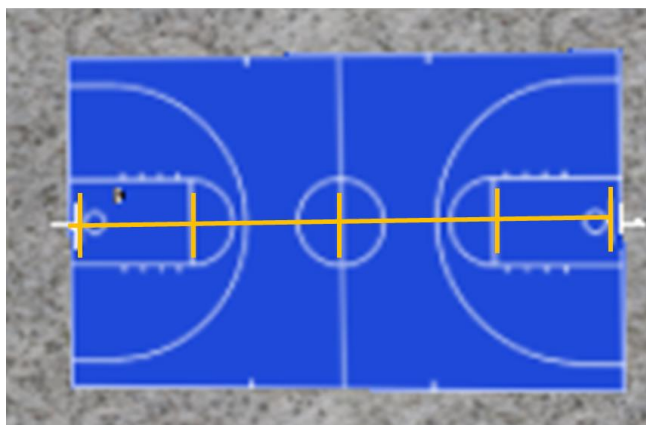


Figura 18. Cancha mostrando distancias que recorre el personaje.

Animación de Carro:

La animación está formada por cinco objetos, la carrocería y cuatro ruedas, considero que lo más complicado de colocar los objetos fue las proporciones y colocar las ruedas en la posición correcta para que no se vean más grandes de lo que deberían. En la carrocería se colocaron las variables que permiten su movimiento en X y Z y una para realizar el giro sobre Y; para las ruedas solo se ocupa una variable para su giro sobre X.

Para que el auto realice sus movimientos se necesita que el usuario presione la tecla **SPACE** para que avance o se detenga. Para el recorrido se decidió que empiece al lado del objeto de entrada y termine estacionado. Lo primero que se debe hacer es verificar en qué estado se encuentra el auto para que avance y giren las llantas hacia una dirección, después se pregunta si ya llego hasta una posición que seleccionamos para que realice un giro, cambie de estado y siga avanzando; las posiciones dependen de la distribución del asfalto. Considero que eso fue lo más

complicado de la animación fue ver hasta qué punto debe girar y no atraviesa objetos que no deseamos.



Animación Hombre que camina:

Para esta animación solo se ocupó un objeto descargado de Mixamo. El personaje se colocó debajo de la entrada y con dos variables para su movimiento en X y Z se desplazará hasta llegar al OXXO y con una variable en su rotación girará sobre el eje Y.

Dentro de la función animate, se colocó un IF para que el usuario presione la tecla 1 y así el personaje empiece su recorrido. Posteriormente se verifica en qué estado se encuentra para que avance en el sentido correcto y al llegar hasta cierta ubicación, éste cambia de orientación y de estado. Cabe aclarar que en el estado cinco, el personaje no avanzará si es que el automóvil va a cruzar por el mismo sitio, esto verificando que el estado del auto sea diferente a dos; a su vez el personaje solo atraviesa el objeto OXXO, no se abren las puertas para que entre.



Animación Persona paseando perro.

Para realizar la animación correspondiente, primero se definió una constante, la cual indicará la velocidad en la que los personajes además de 5 variables flotantes, donde estas nos ayudaran a trasladar los objetos en el eje X, Z y finalmente rotar estos mismos.

Otra variable que también se declara, es la del estado de la animación, la cuál será la que definirá el cambio de estado por medio de un switch-case.

Para que los objetos puedan realizar la animación, es importante que donde dibujamos estos mismos, agreguemos las variables correspondientes en la traslación (X, Z), donde estas se sumarán con su posición definida, de manera que la posición inicial no será modificada. De igual manera, para que los objetos puedan realizar una rotación, se agrega la variable correspondiente en esta.

Para realizar los estados de la animación se utiliza la estructura de control Switch-Case, de manera que cuando el resultado esperado se cumpla, pase a otro caso y continúe la animación.

Para cada caso se encuentra un if-else el cuál va a comprobar que el movimiento en determinado eje no supere al que determinó el desarrollador, lo que consecuencia los objetos aumentarán o disminuirán su posición hasta este límite además de rotar los grados establecidos. Finalmente, una vez la condición no se cumpla, el perro aumentará o disminuirá su posición de manera que este logre que el perro esté al

lado izquierdo de la persona y se pase al siguiente caso (estado de la animación con un máximo de 5 estados).

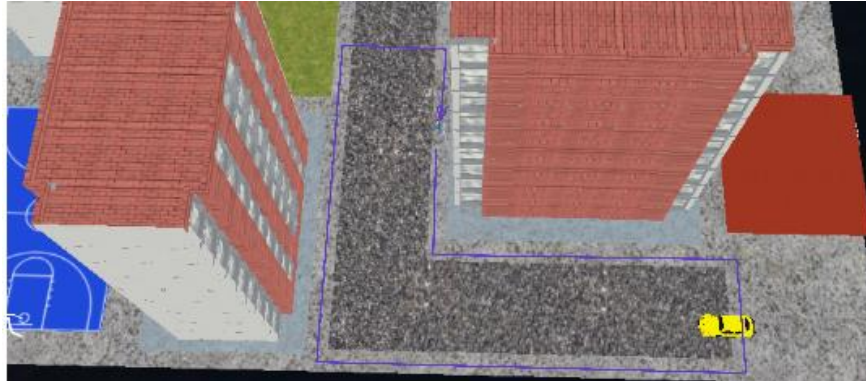


Figura 19. Ruta de la persona paseando un perro.

Animación Silla moviéndose.

Para realizar esta animación lo único que se hace es trasladar en el eje z la silla, para esto creamos una variable global con que va a ir cambiando de valor y con ayuda de una estructura Switch creamos 2 estados lo cuales son los límites de la traslación.

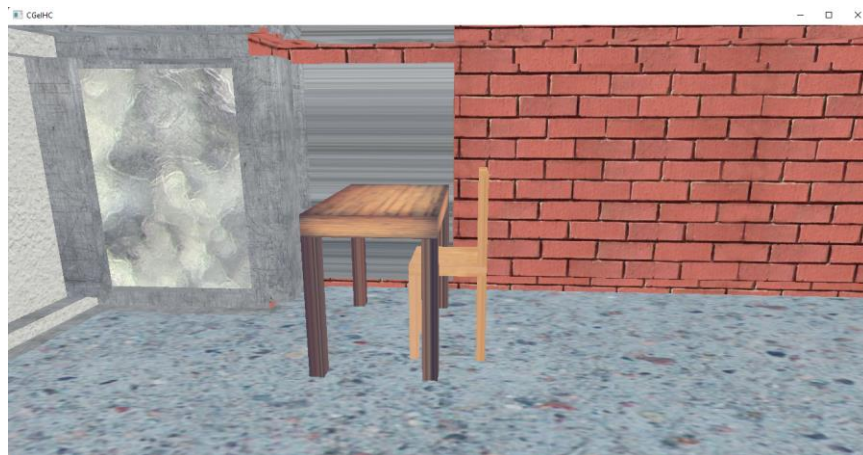


Figura . Silla y mesa.

Animación Refrigerador abriendo la puerta.

Para esta animación lo único que se hace es una rotación en el eje Y de la puerta, al igual que en la silla creamos una variable que va a ir cambiando de valor y dos estados para los límites de la rotación.



Figura 19. Refrigerador.

Animación Ventilador moviéndose.

Esta animación fue un poco mas compleja ya que la cubierta del ventilador rota en el eje Y mientras que las aspas rotan en el eje X, para esto ocupamos dos variables una para cada giro y solo dos estados para los limites de la rotación en Y.

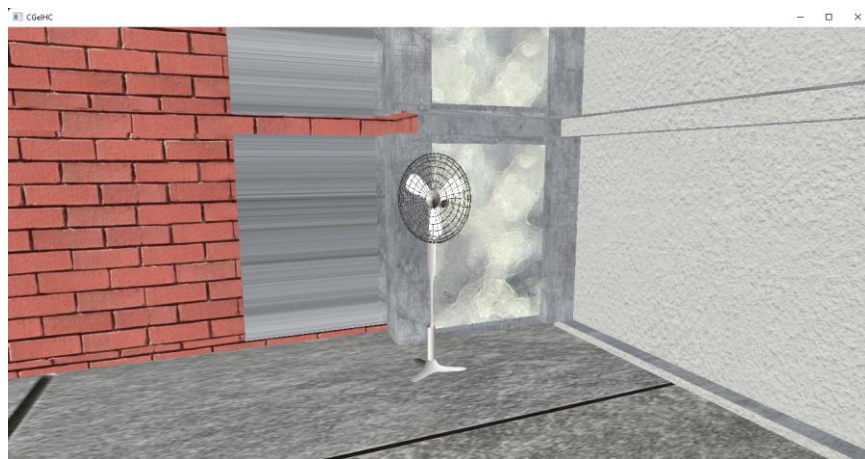


Figura 19. Ventilador.

Estudio técnico para la determinación del costo del proyecto:

El costo del proyecto se divide en dos grandes rubros, el costo directo que está en función de la duración total del proyecto y el costo indirecto que depende de todos los gastos asociados al proyecto. Para hacer el calculo consideremos 4 integrantes en el equipo.

Los cálculos se muestran en las siguientes tablas:

Costo directo	Proyecto final de CGeIHC									
	Cantidad	Personal	Rol	No. de proyectos en los que trabaja	Tarifa/h	Disposición	Horas/Mes	Horas/Proyecto	Salario/Mes	Salario/Proyecto
	1	Líder de proyecto	X	5	\$700.00	0.2	32	64	\$22,400.00	\$44,800.00
	3	Desarrollador	Jr.	5	\$350.00	0.2	32	64	\$11,200.00	\$22,400.00
				6		0.2	32	55	\$11,200.00	\$19,250.00
				7		0.2	32	45	\$11,200.00	\$15,750.00
							Total	256		\$102,200.00

Costo indirecto	Cantidad	Concepto	Costo dólares unitario	Costo pesos unitario	Costo total por concepto
	1	Modelo de automóvil	\$0.00	\$0.00	\$0.00
	5	Modelo de persona	\$0.00	\$0.00	\$0.00
	1	Modelo de perro	\$0.00	\$0.00	\$0.00
	1	Modelo de bicicleta	\$0.00	\$0.00	\$0.00
	0	Servidor	\$0.00	\$0.00	\$0.00
	0	Licencias/Mes	\$244.00	\$4,824.37	\$964.87
	1	Renta de local/Mes	X	\$15,000.00	\$4,500.00
	X	Pago de servicios/Mes	X	\$10,000.00	\$3,000.00
				Total	\$8,464.874

Costo directo:

Para el cálculo del costo directo se consideró una jornada laboral de 8 horas cinco días a la semana, lo que resulta a 40 horas laborales de trabajo; teniendo en cuenta que el proyecto duró 8 semanas, equivalentes a 320 horas.

Duración	horas
1 día	8
1 semana	40
1 mes	160
Proyecto (8S)	320

Dado que somos cuatro integrantes en el equipo, se designó a un miembro como líder del proyecto y el resto con el rol de desarrollador Jr.; cabe aclarar que las tarifas reflejadas en las tablas fueron recuperadas de un caso de estudio en la materia de finanzas. A su vez, como todos los integrantes del equipo llevan más de una materia, su disponibilidad no sería al cien por ciento, por lo cual el rubro de “número de proyectos en lo que se participa” es equivalente a “número de materias que está cursando”. Las columnas siguientes son multiplicaciones hasta el cálculo del total de horas hombre trabajadas aproximado a 256, lo que implica un costo directo de \$102,200.00.

Costo indirecto:

En el costo indirecto se debe considerar todos los elementos utilizados a lo largo del proyecto, como los modelos descargados de internet, los cuales por fortuna no costaron nada, ahí consideramos el rubro de su precio en dólares americanos ya que en las plataformas de descarga su valor se encuentra en este tipo de moneda.

También se debe considerar todos los gastos implícitos, como es la renta de un local, licencias de los diferentes softwares como GitHub Pro, Visual Studio business y 3Ds Max, y el pago de los servicios que se necesitan (internet, electricidad, agua, etc.). Estos gastos de la misma manera se deben distribuir entre todos los proyectos que se realizan por los miembros del equipo como se muestra a continuación, cabe aclarar que se multiplicó entre la disposición de un integrante más grande.

$$\$4,824.37 \cdot 0.2 = \$964.87$$

$$\$15,000.00 \cdot 0.2 = \$4,500.00$$

$$\$10,000.00 \cdot 0.2 = \$3,000.00$$

Lo que resulta en un costo indirecto total de \$8,464.87.

Costo total:

El costo total del proyecto corresponde a la suma de ambos valores totales, el del costo directo y el del indirecto, por lo tanto, se concluye que el costo total del proyecto es igual a \$110,664.87.

Manual de usuario y Mapa de navegación:

https://github.com/Fernando1612/Proyecto-Final-CGEIH-C/blob/316054416_ProyectoFinalTeo_GPO4/README.md

Video explicativo:

<https://youtu.be/Yrzgm1TxxlU>

Limitantes.

Al ambiente virtual le faltaría limitar el recorrido de la cámara para no atravesar los objetos del espacio, principalmente para no posicionarse por debajo del plano, ya que el skybox perdería sentido.

A las animaciones les faltó un poco de complejidad para no ser prácticamente lineales.

Finalmente, el archivo ejecutable únicamente pude ser ejecutado en sistemas Windows y con recursos algo altos para poder mostrar todo el ambiente de una manera correcta.

Actividad Extra

Como actividad extra agregué la biblioteca winmm para reproducir música en nuestro ambiente creando la función de music que se activa cuando se presiona la tecla "Z" y usando la función PlaySound reproduce "TamalesOaxaqueños.wav".

Conclusiones.

La realización de este proyecto fue muy interesante, pero mucho trabajo desde modelar los objetos, agregar texturas, animarlos y finalmente agregarlos al proyecto, en lo personal me tomó más tiempo aprender los diferentes softwares para modelar, editar imágenes y animar. además de programación obviamente considero que un proyecto como este no se puede realizar solo si se quiere obtener un buen resultado, sino que se necesita un equipo de especialistas en modelado, UX/UI, programadores, etc. Este proyecto me ayudó a darme cuenta de todo el trabajo y la inversión detrás de proyectos como los videojuegos y qué habilidades se necesitan para dedicarse a esa área.

Referencias.

- Edificios de México (s.f.). Villa Olímpica Miguel Hidalgo [mensaje en un blog]. Recuperado de <https://www.edemx.com/site/villa-olimpica-miguel-hidalgo/>
- MIXAMO. (2022). Recuperado de <https://www.mixamo.com/#/>
- Open3DModel. (2022). Recuperado de <https://open3dmodel.com/es/>