



MANUAL DE USUARIO.

Proyecto: Feria.



Integrantes:

- **Zagoya Mellado Roberto Uriel**
Número de cuenta: 416113778
- **Orozco Hernández Alexis**
Número de cuenta: 313140255

Asignatura: Laboratorio de Computación Gráfica e interacción humana

Grupo teoría: 04

Grupo Laboratorio: 04

Nombre del profesor: **ING. ARTURO PEREZ DE LA CRUZ**

Escuela: Universidad Nacional Autónoma de México
Facultad de ingeniería.



Índice

Introducción.....	1
Objetivos y especificaciones.....	2
Controles.....	3
Modelos exportados.....	5
Modelos a base de primitivas.....	12
Controles.....	12
Modelos a base de primitivas.....	14
Modelo de montaña rusa.....	15
Modelo de rueda de la fortuna.	16
SkyBox.....	16
Texturas.....	18
Día y noche	19
Animación.	21



Este proyecto trata de la simulación de una feria de pueblo, o parque de diversiones como normalmente lo conocen, se llevó a cabo con el cuidado y principal prioridad de cumplir los alineamientos requeridos para la entrega del mismo y definidos en clase y por los requisitos abordados en las normas vistas en el documento de especificaciones, con el fin de que se pueda comprender este proyecto y presentarse ejemplos claros de cada aplicación, función, modelaje, etc. de esta feria se desarrollara este documento donde se mostrara elementos de la programación como la elaboración de las primitivas, como se exportaron los modelos, funciones que se emplean, etc. como del correcto manejo del programa en ejecución es decir como poder moverse dentro de la feria, ver las animaciones, etc. Aun sin saber de programación grafica ni nada por el estilo, es decir que va orientado a un usuario final que no necesita ser especialista en el área para poder ejecutar y controlar este programa de la feria. También el objetivo de este documento va guiado a un usuario final que si tenga conocimientos de la programación grafica para que en caso de requerirse se pueda dar seguimiento de forma sencilla y por ello se adjuntan capturas de pantalla y ejemplos para facilitar la comprensión de este proyecto. Pero principalmente se realiza para que se pueda observar la elaboración del programa paso a pasa y cómo es que se elaboró y verificar que este cumple con los objetivos. Primero partimos de una breve introducción como se está observando, después procedemos a mostrar los objetivos del proyecto y finalmente la muestra la programación de los modelos exportados, manipulación del programa, animaciones, y la elaboración de los juegos realizados por primitivas, así como sus animaciones, las funciones, iluminación y texturizado de estas últimas, todo esto con descripción y ejemplos de lo que se realizó.





Objetivos y especificaciones.



En esta sección se hace un recordatorio de los objetivos y especificaciones que se acordaron en el documento de especificaciones.

- Se deberá realizar un recorrido virtual de una feria donde se incluirán elementos generados por cada alumno en específico.
- Además de una ambientación tridimensional para simular lo mejor posible el ambiente virtual.
- El proyecto se puede realizar de forma individual o en parejas.
- Se deberán colocar mínimo 2 modelos distintos de juegos mecánicos diferentes realizados a partir de primitivas creadas por ustedes.
- Estos juegos mecánicos deberán ser un tipo montaña rusa y otro tipo giratorio, buscando que sean lo más realista posible.
- Los demás modelos podrán ser cargados como mejor lo consideren.
- Todo el parque deberá estar correctamente texturizado.
- Deberá contener dos modos de visualización.
- Modo de día:
 - Se deberán visualizar los modelos correctamente texturizados sin iluminación.
- Modo de noche:
 - se deberán visualizar los modelos con iluminación y materiales adecuadamente al ambiente nocturno.
- Se generarán 2 opciones para el recorrido, uno para hacerlo de manera libre y otro con un recorrido preestablecido.
- Deberá contener mínimo 2 animaciones adicionales a las de los 2 juegos mecánicos diferentes complejas en el escenario para que sea dinámico.
- El proyecto deberá ser entregado en un disco, donde se colocará el proyecto de VisualStudio, al menos 3 videos diferentes del proyecto funcionando y un manual de usuario.
- El manual de usuario deberá estar en formato .pdf y debe contener capturas de pantalla que ejemplifiquen cada elemento que describen en el manual y al final capturas de pantalla del proyecto.
- Además del disco el proyecto deberá estar en un repositorio exclusivo donde se evaluará cada avance que se vaya realizando en él.
- Estos son los elementos mínimos que debe haber en el escenario, no significa que sean los únicos, cada equipo deberá ambientar el escenario lo mejor posible, se evaluará además la creatividad y originalidad.



Controles.

Los controles del programa en ejecución se definieron al inicio del código como se puede mostrar en la siguiente captura.

```
/*-----*/
/* ----- Proyecto Final -----*/
/*----- 2019-2 -----*/
/*-- Computación gráfica e interacción humano computadora --*/
/*-----Integrantes:-----*/
/*----- Orozco Hernandez Alexis -----*/
/*----- Zagoya Mellado Roberto Uriel-----*/
/*----- Version VS 2017 -----*/
/*-----Grupo de teoria: 04-----*/
/*----- Grupo 04 -----*/

/*-----Controles:
    W: mover hacia delante
    S: Mover hacia atras
    D: Mover hacia la derecha
    A: Mover hacia la izquierda
    P: Poner de noche
    O : poner de dia
    R: Activar la aniacion de la rueda de la fortuna
    T: Activar las tazas
    D: Desactivar las tazas
    K: Activar el carro de comida
    J: Activar los carros chocones
    L: Detiene los carros chocones
    H: Activa los caballos
    V: Activa animacion por keyframes de la montaña */
```

En esta parte se puede observar los controles que manejan el programa en ejecución, esto es gracias a los casos declarados en las líneas posteriores de código.

```
// -----
void my_input(GLFWwindow *window)
{
    if (glfwGetKey(window, GLFW_KEY_ESCAPE) == GLFW_PRESS)
        glfwSetWindowShouldClose(window, true);

    if (glfwGetKey(window, GLFW_KEY_W) == GLFW_PRESS) {
        camera.ProcessKeyboard(FORWARD, (float)deltaTime);
    }
    if (glfwGetKey(window, GLFW_KEY_S) == GLFW_PRESS)
        camera.ProcessKeyboard(BACKWARD, (float)deltaTime);
    if (glfwGetKey(window, GLFW_KEY_A) == GLFW_PRESS)
        camera.ProcessKeyboard(LEFT, (float)deltaTime);
    if (glfwGetKey(window, GLFW_KEY_D) == GLFW_PRESS)
        camera.ProcessKeyboard(RIGHT, (float)deltaTime);
    //if (glfwGetKey(window, GLFW_KEY_P) == GLFW_PRESS)
    // play = true;
    //if (glfwGetKey(window, GLFW_KEY_O) == GLFW_PRESS)
    // play = false;
    /*if (glfwGetKey(window, GLFW_KEY_RIGHT) == GLFW_PRESS)
        direction_right = true;
    if (glfwGetKey(window, GLFW_KEY_LEFT) == GLFW_PRESS)
        direction_right = false;*/
```



```
2046 if (glfwGetKey(window, GLFW_KEY_P) == GLFW_PRESS) { //poner de noche
2047     ambient1 = 0.04f;
2048     ambient2 = 0.04f;
2049     ambient3 = 0.04f;
2050 }
2051 if (glfwGetKey(window, GLFW_KEY_O) == GLFW_PRESS) { //poner de dia
2052     ambient1 = 1.0f;
2053     ambient2 = 1.0f;
2054     ambient3 = 1.0f;
2055 }
2056 if (glfwGetKey(window, GLFW_KEY_R) == GLFW_PRESS) //PARA ACTIVAR LA NIMACION DE LA RUEDA
2057     activate_rueda = true;
2058     //-----para tasas giratorias
2059 if (glfwGetKey(window, GLFW_KEY_T) == GLFW_PRESS) //activar taas
2060     activate_tasas = true;
2061 if (glfwGetKey(window, GLFW_KEY_D) == GLFW_PRESS) //descativar tasas
2062     activate_tasas = false;
2063 if (glfwGetKey(window, GLFW_KEY_K) == GLFW_PRESS) //para activar el carro de comida
2064     activate_cc = true;
2065 if (glfwGetKey(window, GLFW_KEY_J) == GLFW_PRESS) { //para activar los carros chocones
2066     stop_cart1 = false;
2067     stop_cart2 = false;
2068     stop_cart3 = false;
```

```
2102 if (glfwGetKey(window, GLFW_KEY_L) == GLFW_PRESS) { //detiene los carros chocones
2103     stop_cart1 = true;
2104     stop_cart2 = true;
2105     stop_cart3 = true;
2106 }
2107
2108 if (glfwGetKey(window, GLFW_KEY_H) == GLFW_PRESS) { //activa los caballos
2109     activate_caballos = true;
2110 }
2111
2112 //-----para activar la animacion por keyframes-----
2113 if (glfwGetKey(window, GLFW_KEY_V) == GLFW_PRESS) { //activa animacion por keyframes de la montaña
2114     if (activate_coaster == false && (FrameIndex > 1))
2115     {
2116         resetElements();
2117         //First Interpolation
2118         interpolation();
2119
2120         activate_coaster = true;
2121         playIndex = 0;
2122         i_curr_steps = 0;
2123     }
2124     else
2125     {
2126         activate_coaster = false;
2127     }
2128 }
```

Donde como se puede observar dependiendo de la tecla presionada, valida las variables para que dependiendo del valor de cada variable en una función que se evalué, haga una determinada función.



El ratón se controla gracias a la siguiente función y a donde lo muevas va a enfocar la cámara.

```
// glfw: whenever the mouse moves, this callback is called
// -----
void mouse_callback(GLFWwindow* window, double xpos, double ypos)
{
    if (firstMouse)
    {
        lastX = xpos;
        lastY = ypos;
        firstMouse = false;
    }

    double xoffset = xpos - lastX;
    double yoffset = lastY - ypos; // reversed since y-coordinates go from bottom to top

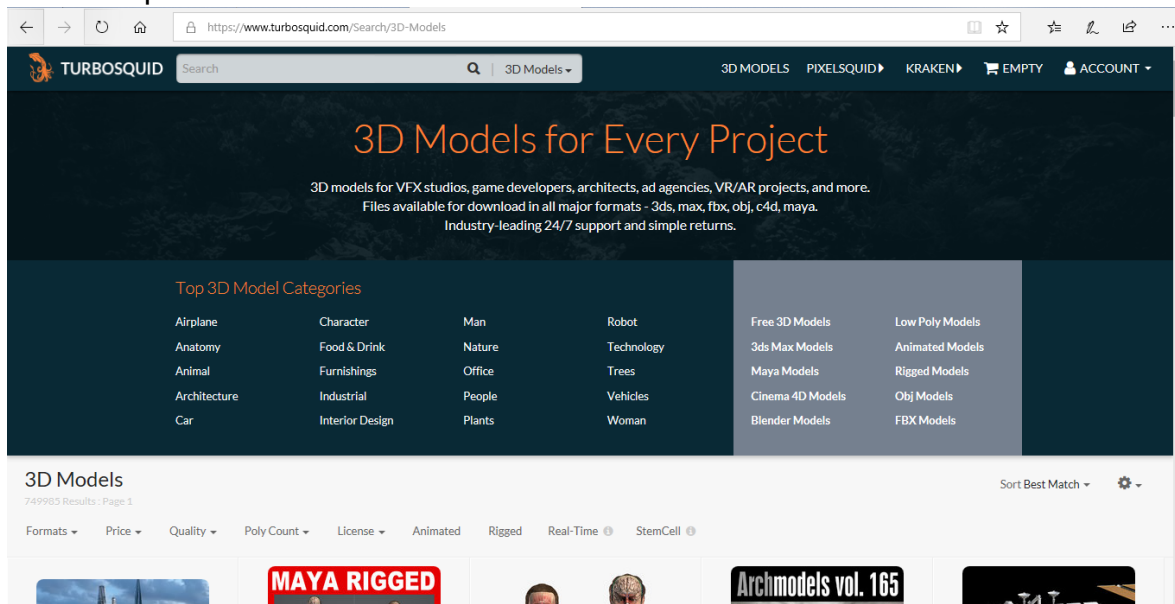
    lastX = xpos;
    lastY = ypos;

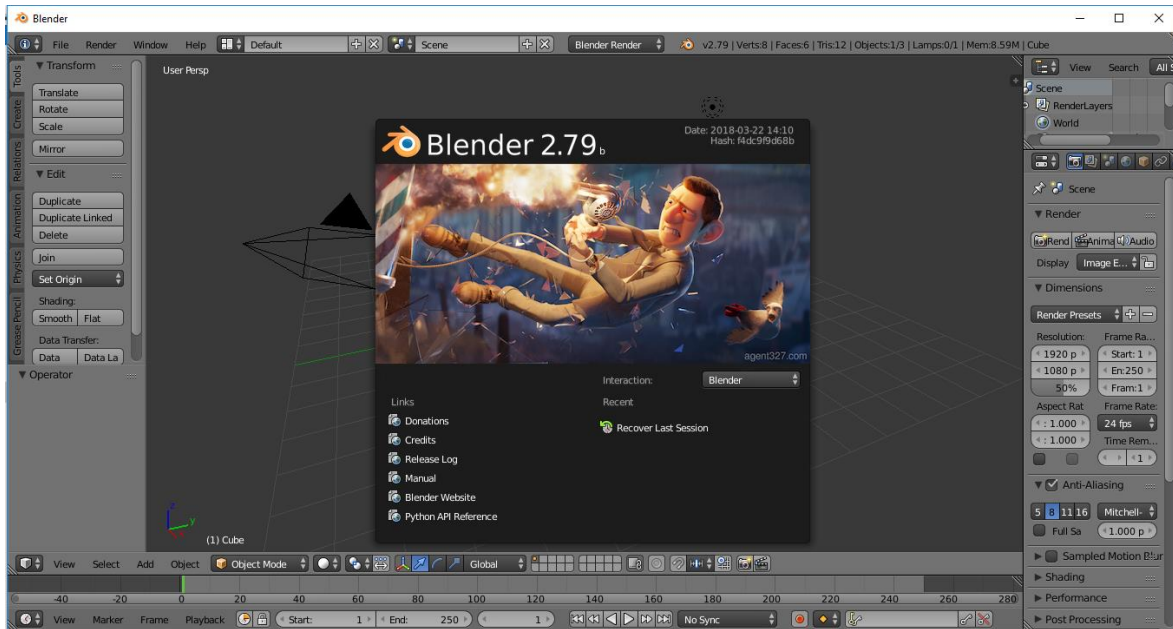
    camera.ProcessMouseMovement(xoffset, yoffset);
}
```

Modelos exportados

En esta sección se encuentran los modelos que se descargaron en la página de internet TurboSquid y posteriormente se colocó la textura en Blender.

TurboSquid





En esta parte se pasan los modelos como parámetros para poder colocarlo de la forma deseada en la feria

```
void display(Shader shader, Shader lampshader, Shader lightingshader, Shader textureshader, Shader lightposshader,
Model carril, Model personaje, Model personaje2, Model futbolito, Model carpa_tasas, Model base_tasas, Model tasa, Model basketball, Model tiro_blanco,
Model jugos, Model ambulante1, Model barrera, Model carpa_fut, Model bumper_car, Model pista, Model silla, Model trash, Model trash2, Model poste_luz1, Model arbol1,
Model premios, Model toilet, Model base_caballos, Model giro_caballos, Model caballo, Model coaster_cart)//ahora si resiven parametros: shader, carril y personaje
```

Los modelos instanciados se declaran de la siguiente forma.

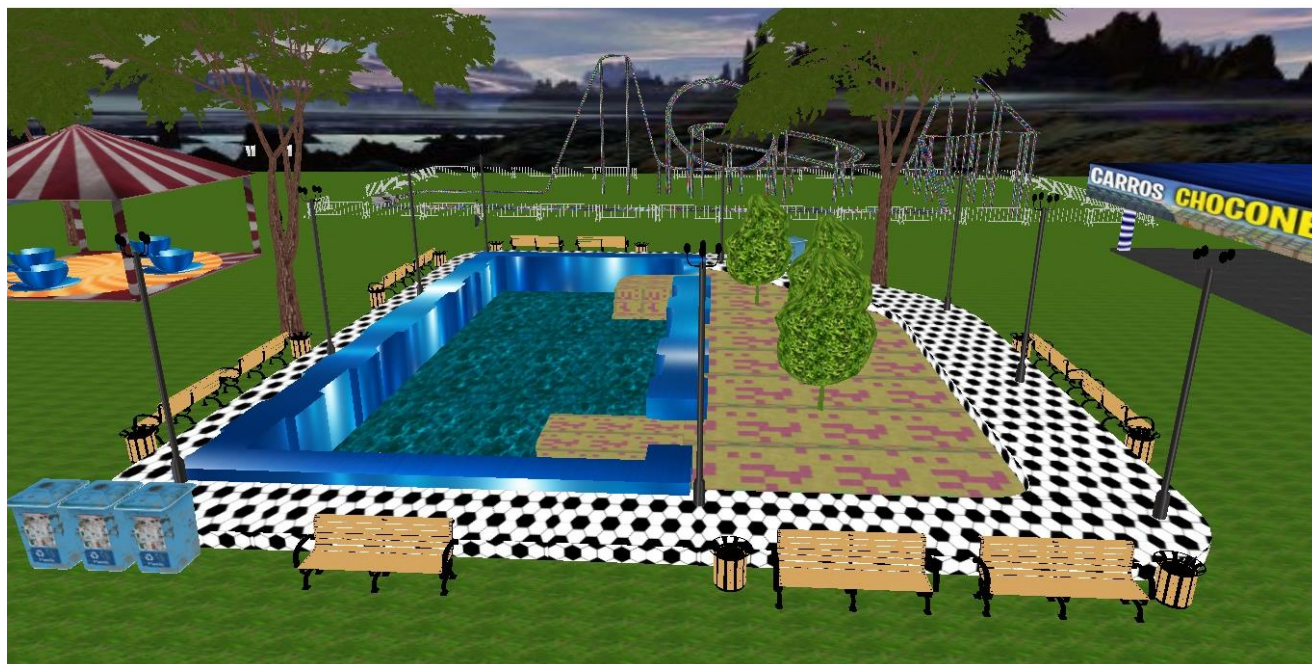
```
// Load models
Model carril = ((char *)"Models/parque/parqueWL.obj");//para cargar el modelo del carril
Model personaje = ((char *)"Models/personaje/muro.obj");//para cargar el modelo del personaje
Model personaje2 = ((char *)"Models/man/man.obj");//para cargar el modelo del personaje
Model carpa_tasas = ((char *)"Models/tasas_giratorias/carpa_tasas.obj");
Model base_tasas = ((char *)"Models/tasas_giratorias/base_giro.obj");
Model tasa = ((char *)"Models/tasas_giratorias/tasa.obj");
Model futbolito = ((char *)"Models/futbolito/futbolito.obj");
Model carpa_fut = ((char *)"Models/futbolito/carpa/carpa_futbolito.obj");
Model basketball = ((char *)"Models/basketball/basketball.obj");
Model jugos = ((char *)"Models/food_carts/jugos/jugos.obj");
//Model jugos = ((char *)"Models/coaster_cart/coaster_cart.obj");

Model ambulante1 = ((char *)"Models/food_carts/jugos/jugos.obj");//carro de comida ambulante
Model barrera = ((char *)"Models/barrera/Barrera.obj");
Model bumper_car = ((char *)"Models/bumper/carrito_chocon.obj");
Model pista = ((char *)"Models/bumper/pista_2.obj");
Model silla = ((char *)"Models/elementos/silla1.obj");
Model poste_luz1 = ((char *)"Models/elementos/poste_luz1.obj");
Model trash = ((char *)"Models/elementos/trash.obj");
Model trash2 = ((char *)"Models/elementos/trash/bote.obj");
Model arbol1 = ((char *)"Models/elementos/arboles/bigtree.obj");
Model tiro_blanco = ((char *)"Models/tiro_blanco/dart_board.obj");
Model premios = ((char *)"Models/juguetes/premios/premios.obj");
Model toilet = ((char *)"Models/toilet/toilet.obj");

Model base_caballos = ((char *)"Models/caballos/base.obj");
Model giro_caballos = ((char *)"Models/caballos/c_giratorio.obj");
```


A continuación, se muestra las imágenes de los modelos empleados en la feria, cabe resaltar que estos modelos son como los hemos estado trabajando en las practicas, es decir se pueden rotar, trasladar y escalar cada modelo, para ponerlo en la posición deseada y la escala deseada de la feria.

Fuente:



cuales se agregaron usando Blender y además cabe resaltar que varios modelos como los arboles dentro de la misma son modelos incluidos, por ello es por lo que aquí no se describe a fondo, pero se pueden verificar en las líneas de código .

Sujeto1





Sujeto2



Tazas giratorias



Como se puede observar este modelo se activa con la tecla T, con lo que realiza su función de girar, esto se debe gracias a la sección de animación. Que como se



menciono anteriormente al recibir un parámetro después de oprimir la tecla T, se activa la animación.

Caballos de feria



De igual forma este modelo realiza la función de girar por medio de animación y esta se activa al oprimir la tecla: H.

Carros chocones



Este modelo cuenta con animación de la cual se activa presionando la tecla J.



Carrito de comida.



Para la animación de este modelo se requiere oprimir la tecla K

Carpas de futbolito





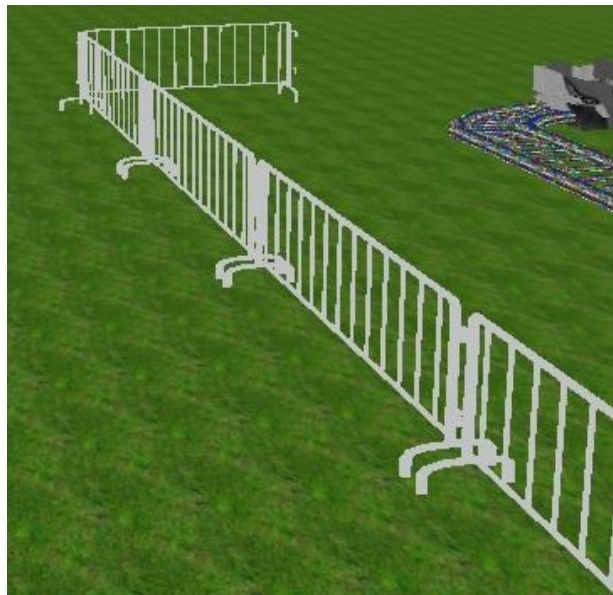
Este modelo no cuenta con animación, sin embargo, cabe resaltar que se realizó a partir de dos modelos que son: el primero la carpa y el segundo las mesas, las cuales simplemente están dibujadas el número de veces que aparecen.

Árbol



De igual forma este modelo no cuenta con animación, pero se compone de un solo modelo.

Reja de seguridad



Estas rejas son las que se encuentran resguardando la montaña rusa, para darle realismo y esta no cuenta con animación pues no es necesaria.



Juegos de baloncesto.



Este modelo no cuenta con animación, pero le da bastante realismo a la feria.

Los modelos anteriores se encuentran en la carpeta de Models que se encuentra dentro del proyecto esto se realiza para poder mandar a llamarlos cuando se utilicen dentro del código.

← → ↕ ⬆					Este equipo > Documentos > unam > 8.-Octavo semestre > 3.-CompuGrafica > Laboratorio > ProyectoFinal > Feria				
					Nombre	Fecha de modifica...	Tipo	Tamaño	
> Acceso rápido					assimp	12/05/2019 11:40 ...	Carpeta de archivos		
> OneDrive					Debug	13/05/2019 05:42 a...	Carpeta de archivos		
v Este equipo					include	12/05/2019 11:40 ...	Carpeta de archivos		
> Descargas					lib	12/05/2019 11:40 ...	Carpeta de archivos		
> Documentos					Models	12/05/2019 11:43 ...	Carpeta de archivos		
> Escritorio					Shaders	12/05/2019 11:40 ...	Carpeta de archivos		
> Imágenes					SOIL2	12/05/2019 11:40 ...	Carpeta de archivos		
> Música					Texturas	12/05/2019 11:43 ...	Carpeta de archivos		
> Objetos 3D					xD	12/05/2019 11:43 ...	Carpeta de archivos		
> Videos					assimp-vc140-mt.dll	12/05/2019 11:15 ...	Extensión de la apl...	15,705 KB	
> Disco local (C:)					camera.h	12/05/2019 11:15 ...	C/C++ Header	4 KB	
> Unidad de CD (E:)					canasta	12/05/2019 11:42 ...	C++ Source File	25 KB	
v URIEL2 (F:)					canasta.h	12/05/2019 11:42 ...	C/C++ Header	1 KB	
> URIEL2 (F:)					cilindro	13/05/2019 12:02 a...	C++ Source File	39 KB	
> 13052019					cilindro.h	13/05/2019 12:02 a...	C/C++ Header	1 KB	
					esfera	12/05/2019 11:15 ...	C++ Source File	4 KB	
					esfera.h	12/05/2019 11:15 ...	C/C++ Header	1 KB	

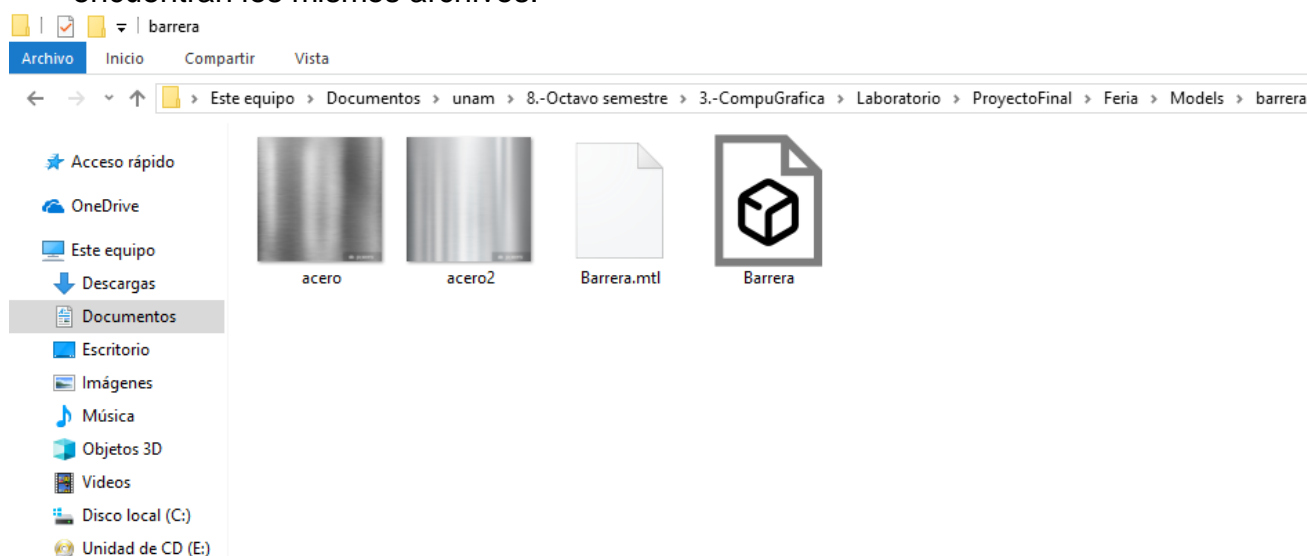


Donde dentro de la carpeta observamos los modelos:

Este equipo > Documentos > unam > 8.-Octavo semestre > 3.-CompuGrafica > Laboratorio > ProyectoFinal > Feria				
	Nombre	Fecha de modifica...	Tipo	Tamaño
do	Arbol	12/05/2019 11:40 ...	Carpeta de archivos	
	barrera	12/05/2019 11:40 ...	Carpeta de archivos	
	basketball	12/05/2019 11:40 ...	Carpeta de archivos	
	bumper	12/05/2019 11:43 ...	Carpeta de archivos	
	caballos	12/05/2019 11:43 ...	Carpeta de archivos	
os	coaster_cart	12/05/2019 11:43 ...	Carpeta de archivos	
	elementos	12/05/2019 11:43 ...	Carpeta de archivos	
	food_carts	12/05/2019 11:43 ...	Carpeta de archivos	
	futbolito	12/05/2019 11:40 ...	Carpeta de archivos	
	hombre	12/05/2019 11:43 ...	Carpeta de archivos	
	juguetes	12/05/2019 11:43 ...	Carpeta de archivos	
	man	12/05/2019 11:43 ...	Carpeta de archivos	
I (C:)	parque	12/05/2019 11:43 ...	Carpeta de archivos	
DVD RW	personaje	12/05/2019 11:40 ...	Carpeta de archivos	
CD (E:)	tasas_giratorias	12/05/2019 11:43 ...	Carpeta de archivos	
	tasas_sin mov	12/05/2019 11:43 ...	Carpeta de archivos	
	tiro_blanco	12/05/2019 11:43 ...	Carpeta de archivos	
	toilet	12/05/2019 11:43 ...	Carpeta de archivos	

Dentro de estas carpetas se encuentran los modelos y las texturas correspondientes a cada una así como el archivo donde se controla cada textura que se usara en el modelo, es importante que se especifique la ruta donde se carga cada modelo, debido a que si no se especifica la ruta y se declara un modelo en una ruta donde ese modelo no existe, causara un error en el código.

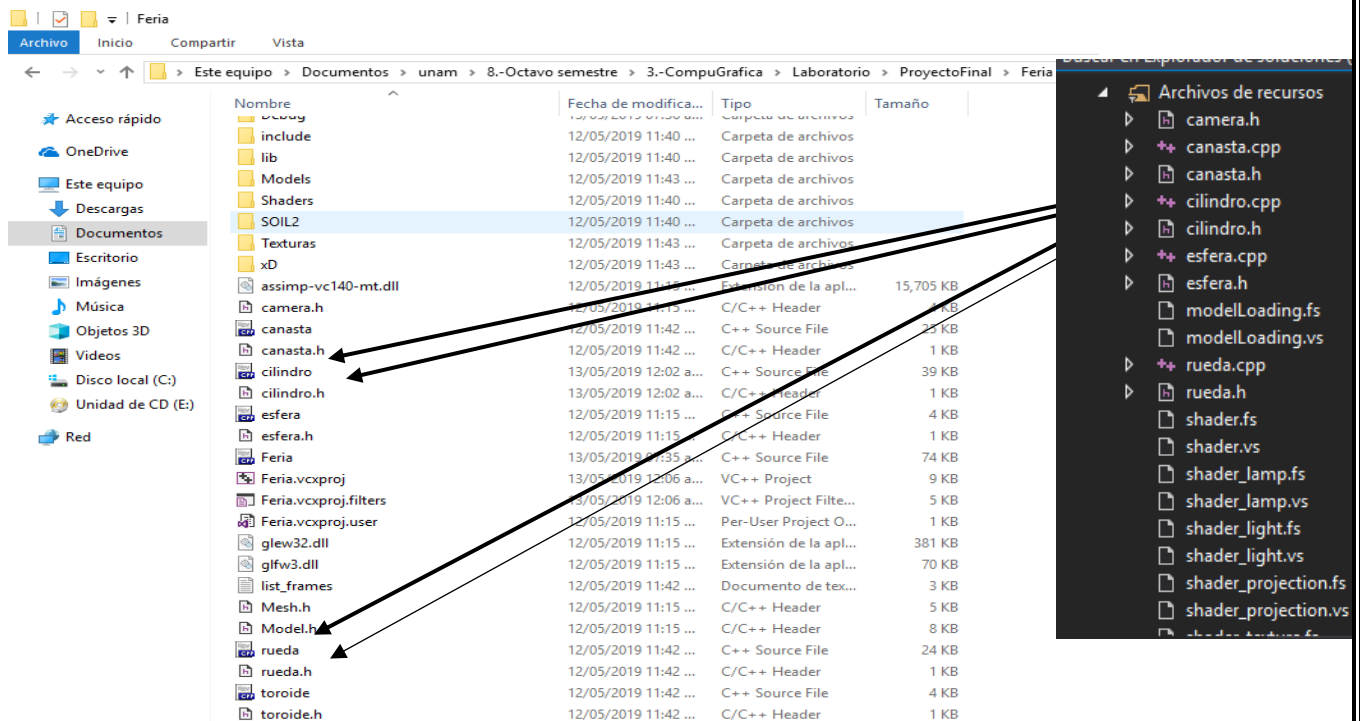
Aquí solo se visualizará dentro de una carpeta de modelo para poder visualizar los archivos correspondientes, pero en general las demás carpetas de modelos se encuentran los mismos archivos.





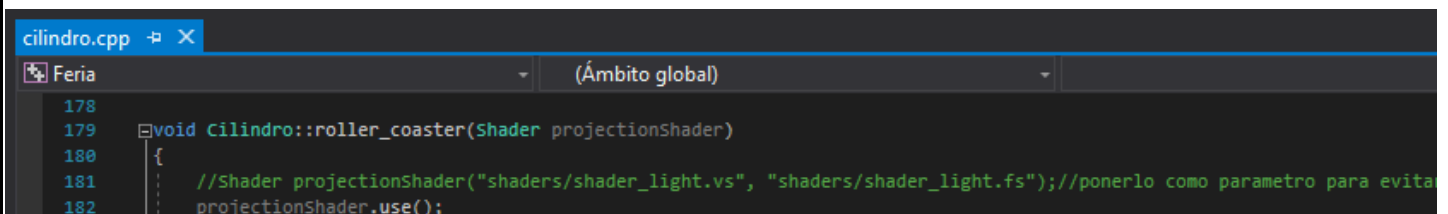
Modelos a base de primitivas.

Como se determinó en las especificaciones este tipo de modelos no se encuentra en carpetas y no simplemente se inicializa, sino que está formado a partir de primitivas y son lo que se conoce como clases que generan objetos y estos objetos simplemente llaman a métodos que son los que se encargan de dibujar o hacer una función en específico, estas “clases” se agregan a los archivos de recursos y se colocan físicamente en una carpeta como se muestra a continuación.



Dentro de estos archivos punto h y punto cpp se encuentran los métodos antes mencionados que se encargan de dibujar estos modelos y así por medio de estas funciones simplifican mucho el código, así como la optimización del programa, y la reutilización de este, a continuación, se muestra los ejemplos de las funciones que realizan las “Clases antes mencionadas”

Por ejemplo, para este caso se encuentra en el archivo de cilindro punto cpp, la función de roller coaster, la cual se encarga de dibujar la montaña rusa.





```
rueda.cpp x cilindro.cpp
Feria (Ámbito global)
265
266 void Rueda::dibujaRueda(unsigned int tex1, Shader lighShader, Shader textureShader, glm::mat4 model, GLuint rueda_VBO, GLuint ru
267 {
268     my_cylinder2.init();
269     my_toroide2.init();
270     Shader projectionShader("shaders/shader_light.vs", "shaders/shader_light.fs");
271     //glBindVertexArray(lightVAO);
272     //glEnable(GL_CULL_FACE);
```

Y en el archivo de rueda.cpp se encuentra el de dibuja rueda que se encarga de dibujar la rueda en el lugar y posición especificada, esta recibe parámetros de

Como se puede observar estas funciones o métodos reciben parámetros los cuales son útiles para poder dibujar el modelo en este caso y para poder optimizar el código, así como la fácil manipulación de estos modelos

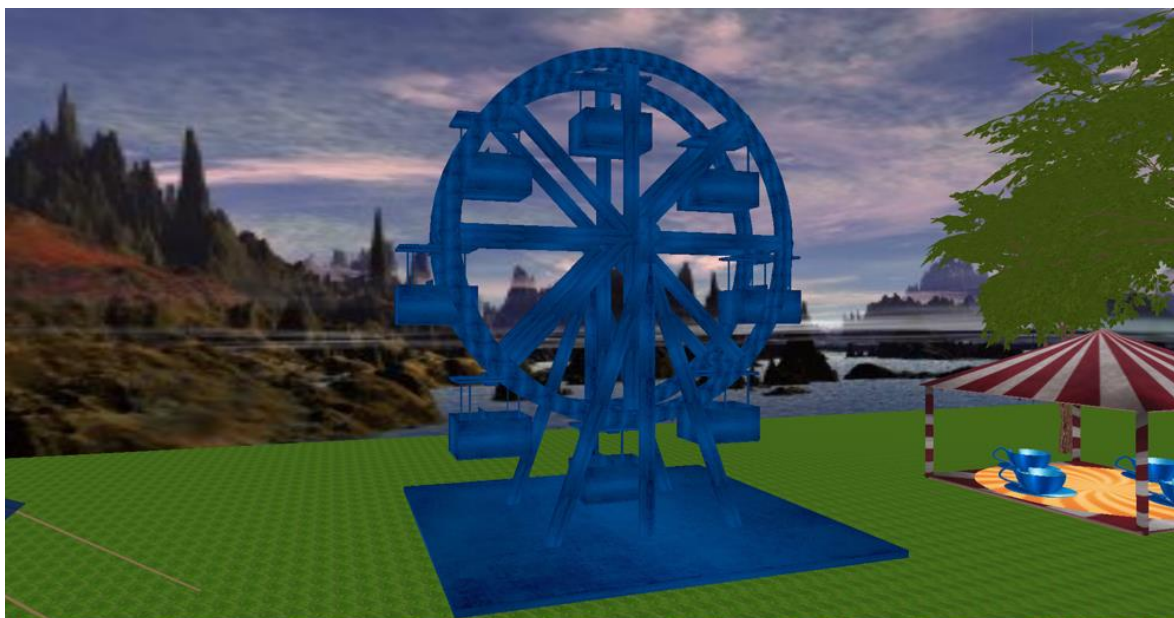
Modelo de montaña rusa.



Este modelo cuenta con animación por key frame y se puede activar con la tecla v, para poder realizar el recorrido sobre la montaña.



Modelo de rueda de la fortuna.



Este modelo activa su animación con la tecla R.

SkyBox

Con el fin de mostrar el sky box de todo el lugar se muestran las siguientes imágenes donde son capturas de pantalla del sky box en ejecución que se muestra desde diferentes ángulos, como se muestra a continuación estas son imágenes que se colocan dentro de una caja de dimensiones grandes superiores al proyecto de la feria,





Practica 9



Dentro del proyecto se pueden mover y explorar a lo largo de las dimensiones del skyBox con las teclas indicadas en el control y el movimiento del mouse, para ver todos los angulos y colocarse dentro del espacio en la posición deseada o a la que se quiera recorrer o visualizar con mas detalle.

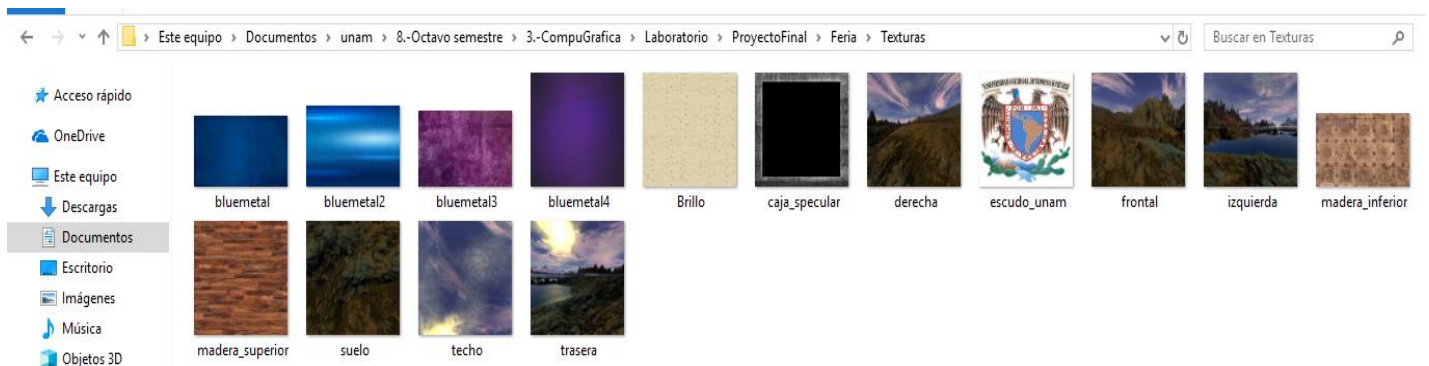
Practica 9





Texturas.

Las texturas que se usan en este proyecto se definen por las de modelos que se encuentran especificadas en la sección de modelos exportados de este documento y las otras que están definidas en las carpetas de texturas.



Para después iniciarlas y utilizarlas adecuadamente estas se usan en elementos de modelos hechos con primitivas y por ejemplo par

```
93 void LoadTextures()  
94 {  
95     t_unam = generateTextures("Texturas/escudo_unam.png", 1);  
96     t_caja_brillo = generateTextures("Texturas/caja_specular.png", 1);  
97     t_coaster = generateTextures("Texturas/bluemetal.jpg", 0);  
98     frontal = generateTextures("Texturas/frontal.jpg", 0);  
99     trasera = generateTextures("Texturas/trasera.jpg", 0);  
100     superior = generateTextures("Texturas/techo.jpg", 0);  
101     inferior = generateTextures("Texturas/suelo.jpg", 0);  
102     derecho = generateTextures("Texturas/derecha.jpg", 0);  
103     izquierdo = generateTextures("Texturas/izquierda.jpg", 0);  
104     brillo = generateTextures("Texturas/brillo.jpg", 0);  
105 }  
106
```

```
glActiveTexture(GL_TEXTURE0);  
glBindTexture(GL_TEXTURE_2D, 0);  
glActiveTexture(GL_TEXTURE1);  
glBindTexture(GL_TEXTURE_2D, t_unam);  
glActiveTexture(GL_TEXTURE2);  
glBindTexture(GL_TEXTURE_2D, t_caja_brillo);  
glActiveTexture(GL_TEXTURE3);  
glBindTexture(GL_TEXTURE_2D, t_coaster);  
glActiveTexture(GL_TEXTURE4);  
glBindTexture(GL_TEXTURE_2D, frontal);  
glActiveTexture(GL_TEXTURE5);  
glBindTexture(GL_TEXTURE_2D, trasera);  
glActiveTexture(GL_TEXTURE6);  
glBindTexture(GL_TEXTURE_2D, superior);  
glActiveTexture(GL_TEXTURE7);  
glBindTexture(GL_TEXTURE_2D, inferior);  
glActiveTexture(GL_TEXTURE8);  
glBindTexture(GL_TEXTURE_2D, derecho);  
glActiveTexture(GL_TEXTURE9);  
glBindTexture(GL_TEXTURE_2D, izquierdo);  
glActiveTexture(GL_TEXTURE10);  
glBindTexture(GL_TEXTURE_2D, brillo);
```




De noche es como se muestra a continuación, se puede apreciar que la iluminación se refleja en las texturas de la rueda de la fortuna y no en el sky box ya que esto le da realismo la luz común que usamos como son lámparas no ilumina todo como lo hace el sol de día, sino hasta su alcance.





Día.



Como se puede observar de día si se ilumina todo por el sol que en nuestra simulación la iluminación ambiental es la que le da el efecto y por medio de animación podemos controlar día y noche con la letra o(amanece) y p(anochece).



Esto se puede observar de mejor forma en el video adjunto o directamente en la ejecución del programa.



Animación.

Como se puede observar los modelos e iluminación contaban con animación, esta se viene trabajando de dos formas en keyframe y en estados, como se vieron en laboratorio a lo largo de las practicas vistas en clase y dependiendo de la tecla que se oprima valida una u otra animación.

```
Feria.cpp x rueda.cpp cilindro.cpp
Feria (Ámbito global)
576
577
578 void animate_horse(void) {
579     if (activate_caballos) {
580         if (vueltas_caballos == 0 || vueltas_caballos == 4) {
581             rot_caballos += 7.0f;
582         }
583         if (vueltas_caballos > 0 && vueltas_caballos < 4) {
584             rot_caballos += 10.0f;
585         }
586         if (vueltas_caballos == 5) {
587             activate_caballos = false;
588             vueltas_caballos = 0;
589         }
590         if (rot_caballos > 360.0) {
591             rot_caballos -= 360.0f;
592             vueltas_caballos += 1;
593         }
594         if (y_caballos < 6.0f && up_caballos) {
595             y_caballos += 0.3f;
596             if (y_caballos >= 6.0f) {
597                 up_caballos = false;
598             }
599         }
600         if (y_caballos > 4.0f && !up_caballos) {
601             y_caballos -= 0.3f;
602             if (y_caballos <= 4.4f) {
603                 up_caballos = true;
604             }
605         }
606     }
607 }
```

Un ejemplo de esto es la función que se definió para animar los caballos en el carrusel, pero, así como esta son varias funciones que se encargan de hacer la animación por el método correcto en el video estas se pueden apreciar mejor o directamente en la ejecución del programa.