315116917, grupo 4

Proyecto Laboratorio de Computación Grafica.

Manual Técnico

Objetivo.

El alumno deberá seleccionar una fachada y un espacio que pueden ser reales o ficticios y presentar imágenes de referencia de dichos espacios para su recreación 3D en OpenGL.

Alcance del proyecto.

Desarrollar un espacio 3D virtual en OpenGL que contenga una ambientación, objetos y animaciones sencillas y complejas que sean lo más parecidas a un entorno de referencia seleccionado por el alumno.

Limitantes.

Para este rubro, me gustaría comentar que tuve dos pequeñas limitaciones, pero que a pesar de ellas pude realizar este proyecto sin mayores inconvenientes.

Estas limitaciones son relacionadas con el carácter adquisitivo y la potencia de cómputo.

La primera está relacionada con que existieron algunos objetos que quería modelar y que los busqué en sitios dedicados a ofrecerlos y eran de opción de pago, misma que no creí recomendable, ya que se me hacia poco práctico invertir dinero en modelos que yo mismo podía crear, por lo que busque modelos similares (en algunas ocasiones más de uno) para modificarlos y que quedaran a como yo los estaba buscando, claro está que esto implicó algo más de tiempo que si sólo hubiera comprado los modelos deseados para usarlos directamente.

La segunda tiene que ver con la capacidad de procesamiento de mi computadora, ya que no dispongo de una laptop que cuente con tarjeta grafica dedicada, lo que influye en que todos los procesos de renderizado de los modelos se ejecutaban en la memoria RAM principal de mi equipo, la cual es de solamente 8GB, al principio no había ningún problema, pero conforme fue incrementando el tamaño del proyecto en cuestión de modelos, a mi quipo le costaba algo de trabajo realizar todos los procesos de renderización y de carga de modelos, además de que al ejecutar de manera simultanea el software de Autodesk Maya y Visual Studio complicaba un poco estas tareas. No me causó muchos inconvenientes, solo debía esperar un poco más de tiempo a que todo se cargara y ejecutara correctamente, lo que dispuso una perdida de tiempo considerable si las sumamos todas al final.

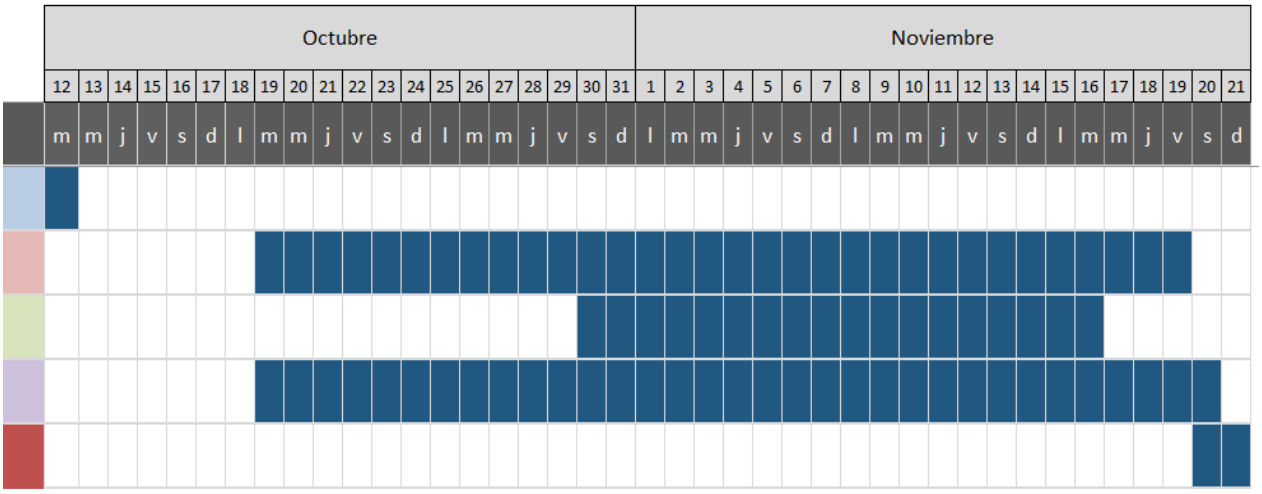
Diagrama de Gantt.

Proyecto Lab. Computación Gráfica

Lab. Computación Gráfica
Nicolas Marin Brian Geovanny

Inicio del proyecto:	mar, 12/10/2021
entrega del proyecto:	sáb, 20/11/2021

TAREA	INICIO	FIN
Selección de espacio a recrear	12/10/2021	12/10/2021
Diseño de modelos en Maya	19/10/2021	19/11/202 1
Generacion de animaciones	30/10/2021	16/11/2021
Carga de todos los elementos en Visual Studio	19/10/2021	20/11/2021
Generación de la documentación	20/11/2021	21/11/2021



Documentación del código.

Diccionario de Variables.

Nombre	Tipo	Función	Tamaño (en bytes)
anim_refri	Booleano	Controla la animación de la puerta de la nevera del refrigerador (puerta de arriba).	1 byte
anim_refri2	Booleano	Controla la animación de la puerta del refrigerador (puerta de abajo).	1 byte
refri_1	Booleano	Activa la abertura de la puerta de la nevera del refrigerador.	1 byte
refri_2	Booleano	Activa el cerrado de la puerta de la nevera del refrigerador.	1 byte
refri_3	Booleano	Activa la abertura de la puerta de abajo del refrigerador.	1 byte
refri_4	Booleano	Activa el cerrado de la puerta de abajo del refrigerador.	1 byte
anim_estufa	Booleano	Controla la animación de la puerta de la estufa.	1 byte
estufa_1	Booleano	Activa la abertura de la puerta de la estufa.	1 byte
estufa_2	Booleano	Activa el cerrado de la puerta de la estufa.	1 byte
anim_entrada	Booleano	Controla la animación de la puerta en la entrada de la casa.	1 byte
puerta_1	Booleano	Activa la abertura de la puerta de la casa.	1 byte
puerta_2	Booleano	Activa el cerrado de la puerta de la casa.	1 byte
anim_cajon	Booleano	Controla la animación del cajón en el tocador.	1 byte
cajon_1	Booleano	Activa la abertura del cajón.	1 byte
cajon_2	Booleano	Activa el cerrado del cajón.	1 byte
anim_elip	Booleano	Controla la animación de la trayectoria del pajarito alrededor de la casa.	1 byte
elipse	Booleano	Controla la primera parte de la trayectoria del pajarito en el plano X a Z.	1 byte
elipse2	Booleano	Controla la segunda parte de la trayectoria del pajarito en el plano Z a X.	1 byte

band	Booleano	Variable que sirve como bandera de control en la primera parte de la trayectoria del pajarito.	1 byte
band2	Booleano	Variable que sirve como bandera de control en la segunda parte de trayectoria del pajarito.	1 byte
band3	Booleano	Variable que sirve como bandera de control al final de la trayectoria del pajarito.	1 byte
alas	Booleano	Controla el aleteo hacia arriba del pajarito.	1 byte
alas2	Booleano	Controla el aleteo hacia abajo del pajarito.	1 byte
anim_caida	Booleano	Controla la animación de la caída de la pelota por las escaleras.	1 byte
pelota	Booleano	Activa el descenso de la pelota a través de las escaleras.	1 byte
aux	Booleano	Variable que sirve como bandera de control en los botes de la pelota en los escalones.	1 byte
aux2	Booleano	Variable que sirve como bandera de control en los botes de la pelota en los escalones.	1 byte
aux3	Booleano	Variable que sirve como bandera de control en los botes de la pelota en los escalones.	1 byte
fin	Booleano	Variable que sirve como bandera de control en el desplazamiento de la pelota contra la pared.	1 byte
rot	flotante	Marca los incrementos y decrementos para la apertura y cerradura de la puerta de la nevera del refrigerador.	4 bytes
rot2	flotante	Marca los incrementos y decrementos para la apertura y cerradura de la puerta de abajo del refrigerador.	4 bytes
rot3	flotante	Marca los incrementos y decrementos para la apertura y cerradura de la puerta de la estufa.	4 bytes
rot4	flotante	Marca los incrementos y decrementos para la apertura y cerradura de la puerta de entrada a la casa.	4 bytes

cajon	flotante	Marca los incrementos y decrementos para la apertura y cerradura del cajón del tocador.	4 bytes
gira	flotante	Delimita el giro del pajarito en la trayectoria en forma de elipse marcada.	4 bytes
PosX	flotante	Marca los incrementos y decrementos en la posición en el eje x durante la trayectoria de vuelo.	4 bytes
PosZ	flotante	Marca los incrementos y decrementos en la posición en el eje z durante la trayectoria de vuelo.	4 bytes
aleteo	flotante	Marca los incrementos y decrementos del movimiento de las alas del pajarito.	4 bytes
caidaY	flotante	Marca los ascensos y descensos de la pelota durante los botes en los escalones.	4 bytes
caidaX	flotante	Marca el recorrido del desplazamiento de la pelota en el eje x durante la caída.	4 bytes
CaidaY2	flotante	Marca el recorrido del desplazamiento en caída libre de la pelota en el eje y durante la caída.	4 bytes
rotPelota	flotante	Genera la rotación (giro) de la pelota durante la caída.	4 bytes

Diccionario de funciones.

Nombre	Función	Descripción
KeyCallback	Permite usar las teclas A,S,W,D para el movimiento de la cámara sintética dentro de la ambientación, así como las teclas seleccionadas para activar y desactivar las animaciones en los objetos	Realiza la llamada de un evento a ejecución mediante el teclado de la computadora. Si el parámetro de devolución de llamada se establece en nulo, la ventana de muestra no notificará a la aplicación sobre los eventos del teclado.

MouseCallback	Permite usar el trackpad o mouse de la computadora para poder cambiar la dirección hacia donde apunta nuestra cámara sintética.	Realiza la llamada de un evento a ejecución mediante el puntero del equipo. Si el parámetro de devolución de llamada se establece en nulo, la ventana de muestra no notificará a la aplicación sobre los eventos del mouse.
DoMovement	Nos ayuda a que los cálculos que utilizamos para hacer las animaciones con las variables en los objetos se ejecuten de una manera rápida y correcta.	Permite ejecutar instrucciones a una velocidad de procesamiento alta.

Secciones de código relevantes:

Para cargar los modelos en formato .obj previamente adaptados:

```
// Carga de modelos con animacion compleja a utilizar
Model bird((char*)"Models/extras/bird.obj");
Model bird1((char*)"Models/extras/bird_ala1.obj");
Model bird2((char*)"Models/extras/bird_ala2.obj");
Model Pelota((char*)"Models/extras/Pelota.obj");

//Carga de modelos generales a utilizar

Model House((char*)"Models/House CatDog/House_CatDog.obj");
Model Puerta((char*)"Models/House CatDog/Puerta_House.obj");
Model Refrigerador((char*)"Models/House CatDog/Refrigerador.obj");
Model PuertaRefri1((char*)"Models/House CatDog/Arriba_Refrigerador.obj");
Model PuertaRefri2((char*)"Models/House CatDog/Abajo_Refrigerador.obj");
Model Estufa((char*)"Models/House CatDog/Estufa.obj");
Model PuertaEstufa((char*)"Models/House CatDog/Puerta_Estufa.obj");
Model PuertaMesita((char*)"Models/House CatDog/Mesita_Puerta.obj");
```

Si el objeto en cuestión a cargar es estático, puede implantarse de la siguiente manera:

```
model = glm::mat4(1);
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
House.Draw(shader);
```

Si contamos con objetos que contienen una animación, las partes que tienen el movimiento deben cargarse y configurarse de manera individual, como por ejemplo:

```
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-4.0f, 0.0f, -4.5f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f)); //para posicionarlo bien
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
Refrigerador.Draw(shader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-4.0f, 0.0f, -4.5f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f)); //para posicionarlo bien
model = glm::rotate(model, glm::radians(-rot), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
PuertaRefri1.Draw(shader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-4.0f, 0.0f, -4.5f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f)); //para posicionarlo bien
model = glm::rotate(model, glm::radians(-rot2), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
PuertaRefri2.Draw(shader);
```

Para generar el movimiento de la cámara sintética dentro de la función DoMovement:

```
void DoMovement( )
{
    // Camera controls
    if ( keys[GLFW_KEY_W] || keys[GLFW_KEY_UP] )
    {
        camera.ProcessKeyboard( FORWARD, deltaTime );
    }

    if ( keys[GLFW_KEY_S] || keys[GLFW_KEY_DOWN] )
    {
        camera.ProcessKeyboard( BACKWARD, deltaTime );
    }

    if ( keys[GLFW_KEY_A] || keys[GLFW_KEY_LEFT] )
    {
        camera.ProcessKeyboard( LEFT, deltaTime );
    }

    if ( keys[GLFW_KEY_D] || keys[GLFW_KEY_RIGHT] )
    {
        camera.ProcessKeyboard( RIGHT, deltaTime );
    }
```

Ejemplo de configuración de variables para realizar una animación sencilla dentro la función DoMovement:

```
if (refri_1) {
    if (rot <= 90) {
        rot += 0.5f;
    }
}

if (refri_2) {
    if (rot >= 0) {
        rot -= 0.5f;
    }
}

if (refri_3) {
    if (rot2 <= 90) {
        rot2 += 0.5f;
    }
}
```

Ejemplo de configuración de variables para realizar una animación compleja dentro la función DoMovement:

```
if (pelota) {
    if (caidaY > -5.35 && aux) {
        caidaY -= 0.1;
    }
    else if (caidaX > -8.4 && fin) { //primer bote
        aux = false;
        caidaX -= 0.025;
        caidaY2 -= 0.1; //0.0635
        rotPelota += 1.0;

        if (caidaX > -2.7 && aux2) { //segundo bote
            if (caidaY < 2) {
                caidaY += 0.15;
            }
        }
        else {
            aux2 = false;
            aux3 = true;
        }

        if (caidaX > -5.1 && aux3) { //tercer bote
            if (caidaY < 8) {
                caidaY += 0.15;
            }
        }
```

```
        if (caidaY2 < -21.3) { //cuarto bote
            if (caidaY < 12) {
                caidaY += 0.15;
            }
        }

        if (caidaY2 < -27.5) { //quinto bote
            if (caidaY < 16) {
                caidaY += 0.15;
            }
        }

        if (caidaY2 < -31.4) {
            if (caidaY < 18.5) {
                caidaY += 0.15;
            }
        }
    }
    else {
        fin = false;
    }

    if (caidaX > -12.25 && fin == false) {
        caidaX -= 0.025;
        rotPelota += 1.0;
    }
}
```

Ejemplo de configuración de variables para activar y desactivar las animaciones dentro de la función KeyCallback.

```
if (keys[GLFW_KEY_I]) //para la puerta de arriba del refri
{
    anim_refri = not anim_refri;
    if (anim_refri) {
        refri_1 = true;
        refri_2 = false;
    }
    else {
        refri_1 = false;
        refri_2 = true;
    }
}

if (keys[GLFW_KEY_K]) //para la puerta de abajo del refri
{
    anim_refri2 = not anim_refri2;
    if (anim_refri2) {
        refri_3 = true;
        refri_4 = false;
    }
    else {
        refri_3 = false;
        refri_4 = true;
    }

}
```

Explicación de las animaciones complejas utilizadas:

1. Animación del pajarito volando alrededor de la casa.

   Para esta animación, utilicé como referencia la trayectoria que dibuja una elipse, esto con la intención de que el pajarito realicé un recorrido de vuelo similar alrededor de la casa, tomando en cuenta que las dimensiones exteriores de ésta no son proporcionales.

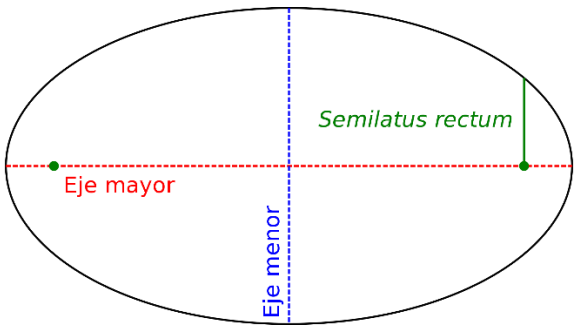
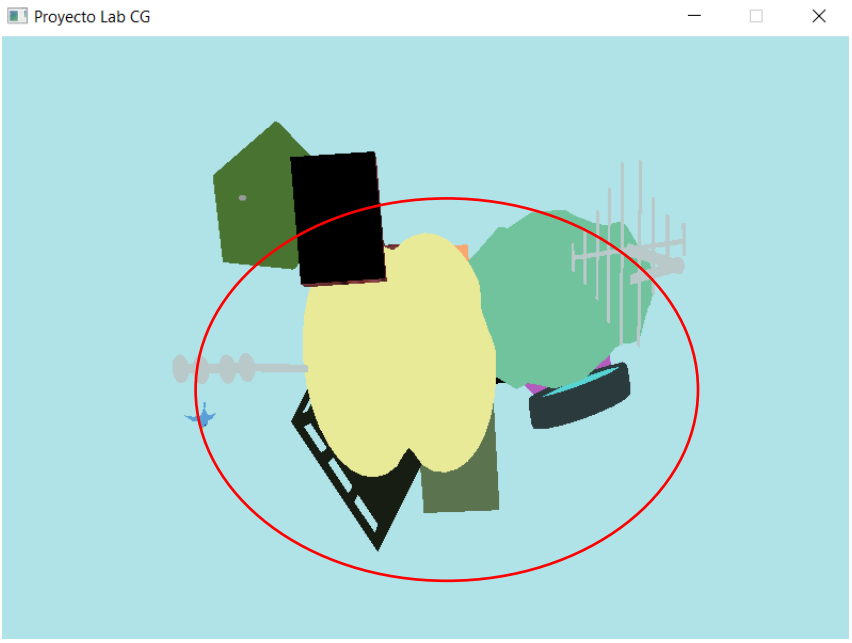


Ilustración 1. Referencia de trayectoria que debe seguir el pajarito

Vista desde arriba de la trayectoria de vuelo del pajarito:



Para diseñar los incrementos y decrementos en las variables alrededor de los ejes X y Z usé como apoyo una tabla de valores de la función de elipse para homologar los valores en el dominio y su respectiva imagen que se deben tener para formar la figura, en este caso la trayectoria del pajarito.

2. Animación de la pelota cayendo por las escaleras.

Para esta animación me base en la trayectoria que forma una parábola negativa y en la de caída libre para simular el descenso de la pelota a través de las escaleras con sus respectivos botes entre ellas.

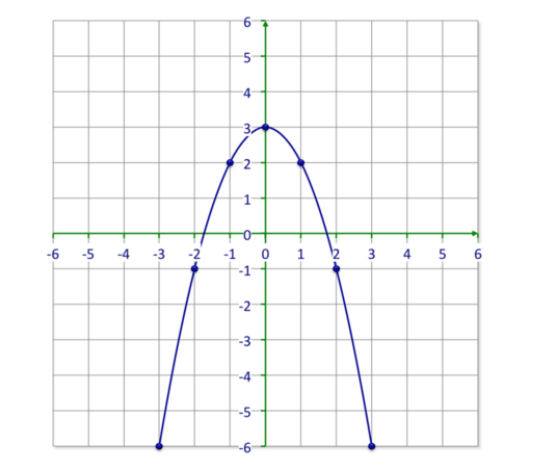
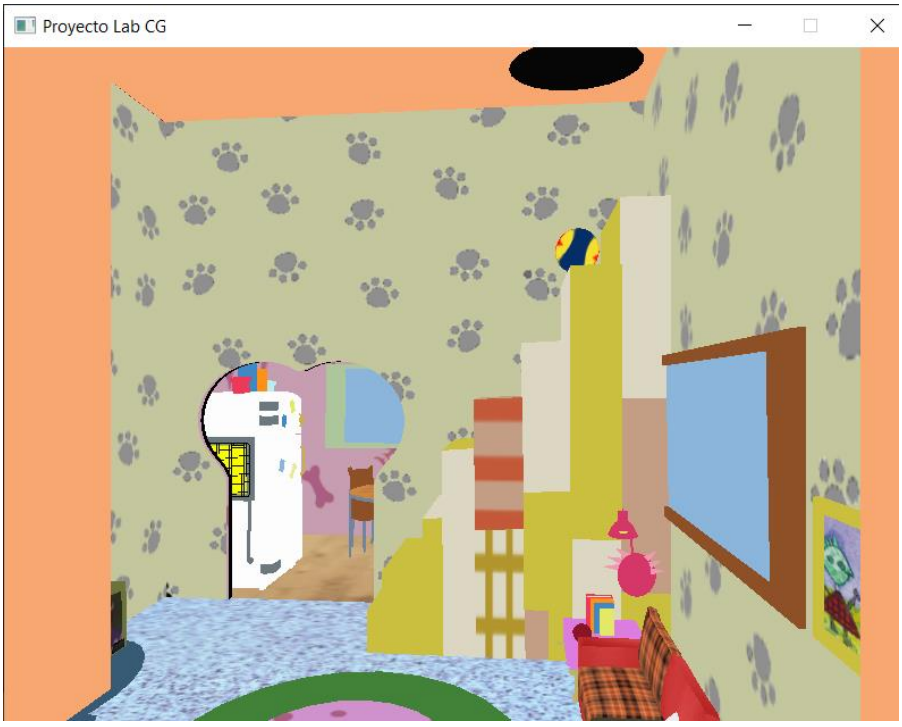


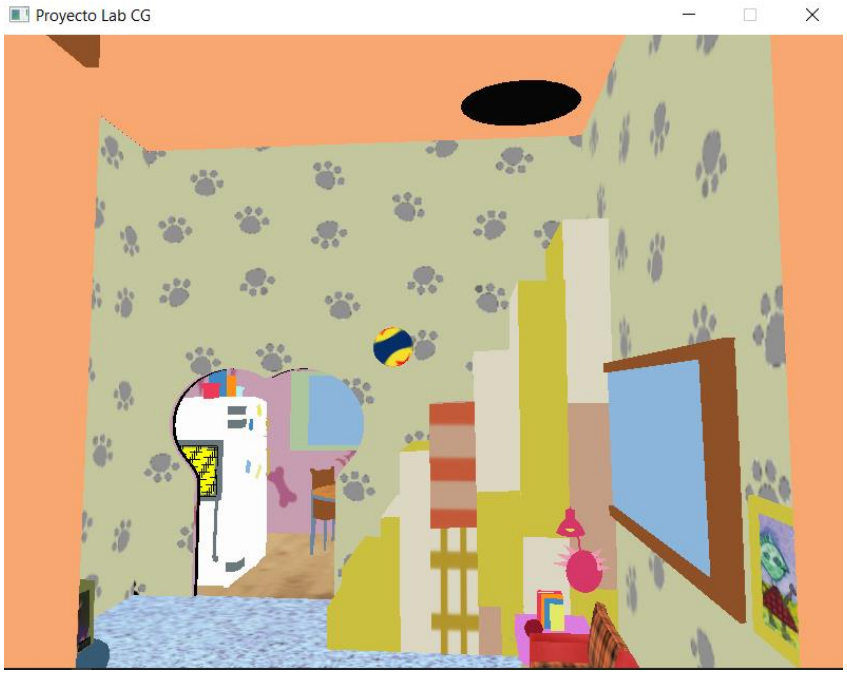
Ilustración 2 y 3. Referencias de trayectoria que debe seguir la pelota al caer

Vista de perfil de la caída de la pelota:




Proyecto Lab CG


Proyecto Lab CG


Proyecto Lab CG


Proyecto Lab CG