Universidad Nacional Autónoma de México

Facultad de Ingeniería

Laboratorio de Computación Gráfica e Interacción Humano-Computadora

Profesor: Carlos Aldair Román Balbuena

Proyecto Final

Manual Técnico

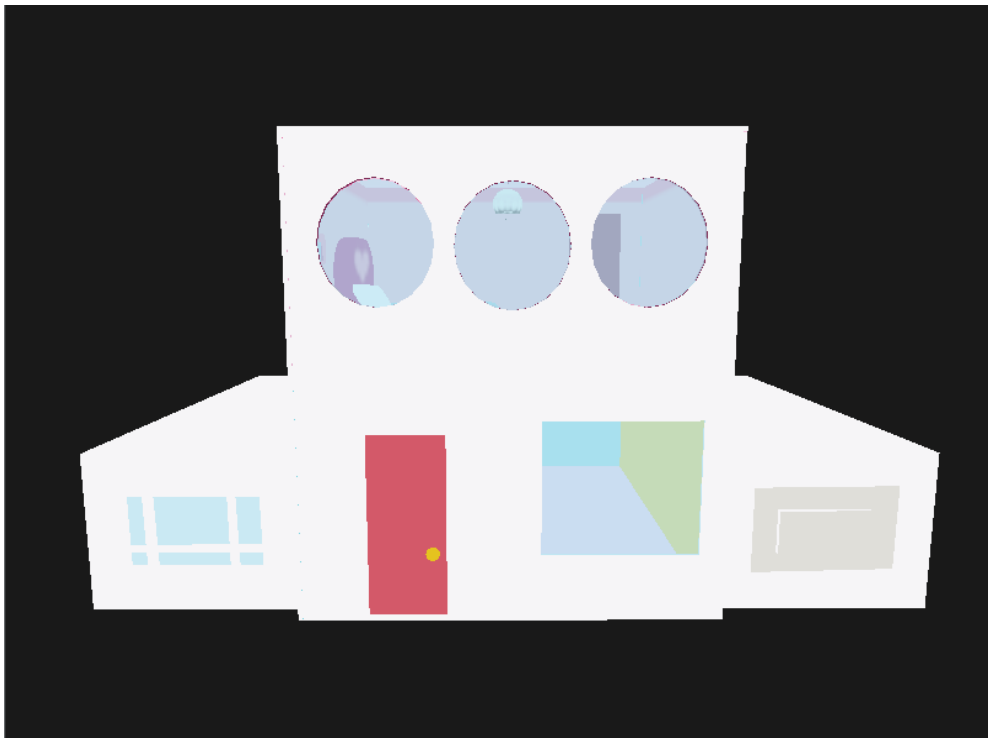
Número de cuenta: 316171580

Grupo: 12

Semestre: 2022-2

Recreación de la casa y habitación de las Chicas Superpoderosas

Manual técnico



Objetivos

- El alumno deberá aplicar y demostrar los conocimientos adquiridos durante todo el curso.

Diagrama de Gantt

Para el Diagrama de Gantt se tomaron en cuenta actividades como el modelado de los diferentes objetos a ocupar, el modelado de la casa y de la habitación principal del proyecto, el maquetado de la casa con los objetos, las animaciones de algunos de los objetos y la realización tanto del manual de usuario como del técnico.

Actividades a realizar	Semana del 11 al 17 de marzo	Semana del 18 al 24 de marzo	Semana del 25 al 31 de marzo	Semana del 1 al 7 de abril	Semana del 8 al 14 de abril	Semana del 15 al 21 de abril	Semana del 22 al 28 de abril	Semana del 29 de abril al 5 de mayo	Semana del 6 al 12 de mayo
Subir imagen de referencia									
Modelar objeto 1									
Modelar objeto 2									
Modelar objeto 3									
Modelar objeto 4									
Modelar objeto 5									
Modelar objeto 6									
Modelar objeto 7									
Modelar cuarto y fachada									
Modelar sala									
Maquetar casa con objetos									

Animar puertas y ventanas									
Animar cajones de mueble									
Animar pelota									
Animar oso de peluche									
Realizar manual de usuario									
Realizar manual técnico									

Alcance del proyecto

En el presente proyecto se busca recrear un espacio de alguna caricatura o de la vida real, para aplicar lo aprendido durante el curso del Laboratorio de Computación Gráfica, por ello se debe de manejar adecuadamente el modelado de objetos, dando a cada uno un texturizado correcto, y manejando los materiales adecuados para que los objetos luzcan lo más parecido a cómo se muestran en la imagen de referencia dada, lo anterior debe ir acompañado de una iluminación apropiada para dar el efecto buscado.

Dentro del desarrollo del proyecto también fueron indispensables las animaciones de los objetos, ya que fue uno de los temas más importantes del curso, para esta parte se desarrollaron tanto animaciones sencillas como complejas. En las sencillas se usaron únicamente las transformaciones básicas, mientras que en las complicadas se usaron combinaciones de transformaciones definidas por condiciones para generar un movimiento específico.

En el caso de este proyecto la casa a recrear fue la de las Chicas Superpoderosas, por ello el producto final debe ser parecido a la caricatura, y debe ir acorde a la misma, tanto en los objetos modelados, como en la estructura de la casa.

Limitantes

Una de las limitantes que se presentó fue la calidad de los texturizados de los objetos, ya que en algunos no se logró darles un aspecto tan definido, esto porque no se logró ocupar imágenes vectorizadas, por lo que la calidad de las imágenes es algo baja, y por ello no se logra quitar el fondo totalmente, ni tener líneas definidas.

Documentación del código

- **Carga de modelos**

Para poder visualizar cada uno de los objetos modelados fue necesario cargarlos en OpenGL.

```
//Carga de modelos
Model casa((char*)"Models/Casa/casaChicas.obj");
Model telefono((char*)"Models/Telefono/Phone2.obj");
Model telefonoArriba((char*)"Models/Telefono/Phone1.obj");
Model mesita((char*)"Models/Mesita/mesitaConTextura.obj");
Model cama((char*)"Models/Cama/Bed3.obj");
Model alfombra((char*)"Models/Alfombra2/alfombra2.obj");
Model lampara((char*)"Models/lampara2/lampara2.obj");
Model repisa((char*)"Models/Repisa/repisaLibros.obj");
Model puertaCasa((char*)"Models/Casa/puerta_casa.obj");
Model puertaCuarto((char*)"Models/Casa/puerta_cuarto.obj");
Model ventana((char*)"Models/Casa/ventana.obj");
Model ventanasCirculares((char*)"Models/Casa/ventana cirular.obj");
Model ventanaConBarrotes((char*)"Models/Casa/ventana con barrotes.obj");
Model barrotes((char*)"Models/Casa/barrotes.obj");
Model mueble((char*)"Models/Mueble/mueble.obj");
Model cajon1((char*)"Models/Mueble/cajon1.obj");
Model cajon2((char*)"Models/Mueble/cajon2.obj");
Model pelota((char*)"Models/Pelota/pelota.obj");
Model oso((char*)"Models/Oso de peluche/oso.obj");
```

- **Luces del ambiente desarrollado**

Se definieron las características de la luz que iba a iluminar el ambiente.

```
// Directional light
glUniform3f(glGetUniformLocation(lightingShader.Program, "dirLight.direction"), -0.2f, -1.0f, -0.3f);
glUniform3f(glGetUniformLocation(lightingShader.Program, "dirLight.ambient"), 0.8f, 0.8f, 0.8f);
glUniform3f(glGetUniformLocation(lightingShader.Program, "dirLight.diffuse"), 1.0f, 1.0f, 1.0f);
glUniform3f(glGetUniformLocation(lightingShader.Program, "dirLight.specular"), 1.0f, 1.0f, 1.0f);
```

- **Se dibujan los objetos en el ambiente de desarrollo**

Dentro de esta sección se muestra cómo se manda a dibujar a los diferentes objetos, en donde se tienen algunos casos en los que se requirió de las transformaciones básicas. Algunos objetos se tuvieron que escalar para ir de acuerdo al tamaño de la casa y de los demás objetos, a la mayoría se les tuvo que aplicar la traslación para posicionarlos en el lugar correspondiente, y otros se tuvieron que rotar para que estuvieran dirigidos a cierta parte de la casa.

En la siguiente imagen se pueden observar las características de algunos de los objetos, dentro de los cuales están la casa, el teléfono y la parte de arriba del teléfono.

La casa era el objeto más importante, ya que era la base para poder posicionar todos los demás objetos, por lo mismo no se le aplicó ningún tipo de transformación.

El teléfono se tuvo que separar en dos partes, pues era uno de los objetos a los que se les daría alguna animación y para mover la parte de la bocina se debía mantener separado. A pesar de eso, a ambas partes se les dieron las mismas transformaciones de escala y de traslación.

A la parte de arriba también se le aplicó una rotación, con la que se moverá la bocina como si el teléfono estuviera recibiendo una llamada.

También es importante mencionar que como el teléfono tiene ojos y boca se requirieron imágenes a las que se les tenía que eliminar el fondo, por ello a la hora de dibujarlo se activa la transparencia.

```
//Objetos
//Casa
model = glm::mat4(1);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
casa.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Telefono
model = glm::mat4(1);
model = glm::scale(model, glm::vec3(0.2f));
model = glm::translate(model, glm::vec3(39.5f, 87.0f, 51.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 1);
telefono.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);

//Telefono Arriba
model = glm::mat4(1);
model = glm::scale(model, glm::vec3(0.2f));
model = glm::translate(model, glm::vec3(39.5f, 87.0f, 51.0f));
model = glm::rotate(model, glm::radians(rotTelefonoArriba), glm::vec3(0.0f, 0.0f, 1.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
telefonoArriba.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
```

Los siguientes objetos que se muestran son la mesita sobre la que está el teléfono, la cama y la alfombra.

Para la mesita sólo se tuvo que aplicar una traslación, ya que el tamaño con el que contaba era adecuado.

En el caso de la cama se escaló, se trasladó y se giró para posicionarla en su lugar correspondiente.

La alfombra únicamente se escaló, pero para poder dibujarla también fue necesario activar la transparencia, pues se tenía que eliminar el fondo de la imagen que se usó para su creación, aunque hay que señalar que se tuvo un poco de problemas con la definición de la imagen, por lo que la parte del fondo que está pegada a la alfombra se quedó sin ser eliminada y en momentos se puede notar.

```
//Mesita
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(8.0f, 15.15f, 10.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
mesita.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Cama
model = glm::mat4(1);
model = glm::scale(model, glm::vec3(1.2f, 1.4f, 1.0f));
model = glm::translate(model, glm::vec3(0.0f, 10.5f, 5.3f));
model = glm::rotate(model, glm::radians(-90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
cama.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Alfombra
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(0.0f, 13.5f, -3.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 1);
alfombra.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
```

En la siguiente imagen se tiene lo correspondiente para la lámpara, la repisa y la puerta de la entrada de la casa.

Para la lámpara también se requirió activar la transparencia, pero sucedió lo mismo que con la alfombra, pues no se logró eliminar el fondo completamente. A este objeto se le aplicó la traslación y la rotación.

A la repisa con libros se le aplicaron las tres transformaciones. Mientras que, la puerta solo fue escalada y trasladada.

```
//Lampara
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(0.0f, 23.55f, 0.0f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 1);
lampara.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);

//Repisa
model = glm::mat4(1);
model = glm::scale(model, glm::vec3(1.8f, 1.8f, 0.7f));
model = glm::translate(model, glm::vec3(4.0f, 11.3f, 16.2f));
model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
repisa.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Puerta entrada casa
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(10.82f, 5.33f, 7.69f));
model = glm::rotate(model, glm::radians(rotPuertaFachada), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
puertaCasa.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
```

Los siguientes modelos que se tienen son la puerta del cuarto, los barrotes de la ventana del lado izquierdo de la casa, el mueble de la habitación y uno de los

cajones del mueble. En general a todos se les aplicó alguna de las transformaciones y no se requirió activar la transparencia.

```
//Puerta cuarto
model = glm::mat4(1);
model = glm::scale(model, glm::vec3(1.0f, 1.01f, 1.0f));
model = glm::translate(model, glm::vec3(-11.2f, 18.53f, -9.78f));
model = glm::rotate(model, glm::radians(rotPuertaCuarto), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
puertaCuarto.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Barrotes
model = glm::mat4(1);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
barrotes.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Mueble
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-0.75f, 14.39f, 5.2f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
mueble.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Cajón1 derecha
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(posicionCajon1, 14.39f, 3.5f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
cajon1.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
```

En esta sección se tiene el otro cajón y un oso de peluche, en ambos casos sucede lo mismo que en los objetos anteriores.

```
//Cajón2 izquierda
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-6.45f, 14.39f, posicionCajon2));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
cajon2.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Oso
model = glm::mat4(1);
model = glm::scale(model, glm::vec3(0.8f));
model = glm::translate(model, glm::vec3(-12.8f, movOsoY, 1.9f));
model = glm::rotate(model, glm::radians(rotOso), glm::vec3(1.0f, 0.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
oso.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
```

Después de estos objetos se tienen aquellos en los que se tenía que activar el canal alfa, pues se requería que se tuviera cierta translucidez, ya que se trataba de las ventanas que tiene la casa.

```
glEnable(GL_BLEND);//Activa la funcionalidad para trabajar el canal alfa
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);

//Ventana fachada
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(10.7f, 9.1f, -1.65f));
model = glm::rotate(model, glm::radians(rot), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
glUniform4f(glGetUniformLocation(lightingShader.Program, "colorAlpha"), 1.0, 1.0, 1.0, 0.85);
ventana.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Ventanas circulares
model = glm::mat4(1);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
glUniform4f(glGetUniformLocation(lightingShader.Program, "colorAlpha"), 1.0, 1.0, 1.0, 0.85);
ventanasCirculares.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Ventana con barrotes
model = glm::mat4(1);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
glUniform4f(glGetUniformLocation(lightingShader.Program, "colorAlpha"), 1.0, 1.0, 1.0, 0.85);
ventanaConBarrotes.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
```

Otro objeto al que se le agregó lo de translucidez fue la pelota, la cual recibió las tres transformaciones básicas.

```
//Pelota
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(movPelotaX, movPelotaY, movPelotaZ));
model = glm::rotate(model, glm::radians(45.0f), glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::rotate(model, glm::radians(rotPelota), glm::vec3(0.0f, 0.0f, 1.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
glUniform4f(glGetUniformLocation(lightingShader.Program, "colorAlpha"), 1.0, 1.0, 1.0, 0.8);
pelota.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));


glDisable(GL_BLEND);  //Desactiva el canal alfa
glUniform4f(glGetUniformLocation(lightingShader.Program, "colorAlpha"), 1.0, 1.0, 1.0, 1.0);
glBindVertexArray(0);
```

Después de poner las características de todos los objetos se relacionaron los objetos que tienen animaciones con las diferentes teclas, esto para indicar cuando se quisiera iniciar o terminar su movimiento. A continuación se muestra un ejemplo de algunas:

```
//Para animación de objetos
if (keys[GLFW_KEY_R]) {
    animVentana1 = true;
}

if (keys[GLFW_KEY_F]) {
    animVentana2 = true;
}

if (keys[GLFW_KEY_T]) {
    animPuertaFachada1 = true;
}

if (keys[GLFW_KEY_G]) {
    animPuertaFachada2 = true;
}

if (keys[GLFW_KEY_Y]) {
    animPuertaCuarto1 = true;
}
```

Esa misma función se usa para dar movimiento a algunos objetos en los que se aplican animaciones sencillas, en estas animaciones se realiza o una u otra transformación de las básicas.

```
if (keys[GLFW_KEY_R])
{
    if (animVentana1) {
        while (rot < 90) {
            rot += 0.1;
        }

    }
}

if (keys[GLFW_KEY_F])
{
    if (animVentana2) {
        while (rot > 0) {
            rot -= 0.1;
        }
    }
}

if (keys[GLFW_KEY_T])
{
    if (animPuertaFachada1) {
        while (rotPuertaFachada < 90) {
            rotPuertaFachada += 0.1;
        }

    }
}
```

Luego se tiene otra función llamada animación() y dentro de la cual se tienen las animaciones complejas.

Esas animaciones son complejas porque se programaron como una máquina de estados, tanto para la pelota, como para el teléfono se hicieron varios casos para determinar la posición del objeto en un estado.

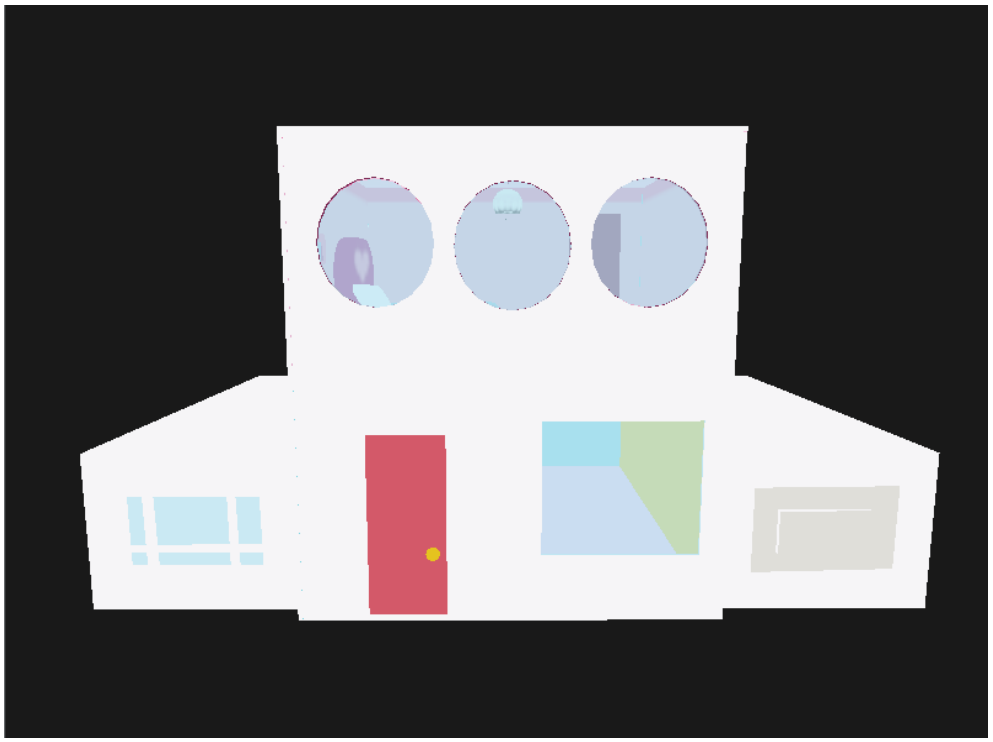
```
void animacion(){
    if (pelotaEnMovimiento)
    {
        if (movPelota1)
        {
            movPelotaY += 0.1f;
            if (movPelotaY > 17.4f) {
                movPelota1 = false;
                movPelota2 = true;
            }
        }
        if (movPelota2)
        {
            movPelotaY -= 0.1f;
            movPelotaX -= 0.05f;
            if (movPelotaY < 14.5f && movPelotaX < 4.5f)
            {
                movPelota2 = false;
                movPelota3 = true;
            }
        }
        if (movPelota3)
        {
            movPelotaY += 0.2f;
            movPelotaX -= 0.1f;
            if (movPelotaY > 16.0f && movPelotaX < 3.0f)
            {
                movPelota3 = false;
                movPelota4 = true;
            }
        }
```

Conclusiones

Con la realización de este proyecto fue posible comprender a mayores rasgos cómo es que funciona la computación gráfica, qué aspectos influyen en el desarrollo de ambientes gráficos y cuál es la importancia de saber usar las diferentes herramientas que tenemos a nuestro alcance para lograr mayores resultados en la realización de proyectos de este tipo. Me parece que es un área de la computación que abarca muchos conocimientos, pero los resultados del desarrollo de ambientes es bastante satisfactorio. En lo personal me gustó mucho realizar esta recreación de la casa, aunque tengo que aceptar que es un área en donde se requiere de mucho trabajo, estudio, concentración, pero sobre todo tiempo.

Recreation of the house and room of the Powerpuff Girls

Technical manual



Objectives

- The student must apply and demonstrate the knowledge acquired throughout the course.

Gantt diagram

For the Gantt Chart, activities were taken into account, such as the modeling of the different objects to be occupied, the modeling of the house and the main room of the project, the layout of the house with the objects, the animations of some of the objects and the realization of both the user manual and the technical one.

Actividades a realizar	Semana del 11 al 17 de marzo	Semana del 18 al 24 de marzo	Semana del 25 al 31 de marzo	Semana del 1 al 7 de abril	Semana del 8 al 14 de abril	Semana del 15 al 21 de abril	Semana del 22 al 28 de abril	Semana del 29 de abril al 5 de mayo	Semana del 6 al 12 de mayo
Subir imagen de referencia									
Modelar objeto 1									
Modelar objeto 2									
Modelar objeto 3									
Modelar objeto 4									
Modelar objeto 5									
Modelar objeto 6									
Modelar objeto 7									
Modelar cuarto y fachada									
Modelar sala									
Maquetar casa con objetos									

Animar puertas y ventanas									
Animar cajones de mueble									
Animar pelota									
Animar oso de peluche									
Realizar manual de usuario									
Realizar manual técnico									

Project scope

This project seeks to recreate a space from a cartoon or from real life, to apply what was learned during the course of the Computer Graphics Laboratory, for this reason the modeling of objects must be properly handled, giving each one a correct texturing and handling the appropriate materials so that the objects look as close as possible to what is shown in the given reference image, the above must be accompanied by appropriate lighting to give the desired effect.

Within the development of the project, the animations of the objects were also essential, since it was one of the most important topics of the course, for this part both simple and complex animations were developed. In the simple ones only the basic transformations were used, while in the complicated ones combinations of transformations defined by conditions were used to generate a specific movement.

In the case of this project, the house to be recreated was that of the Powerpuff Girls, so the final product must be similar to the cartoon, and must be consistent with it, both in modeled objects and in the structure of the house.

Limitations

One of the limitations that was presented was the quality of the texturing of the objects, since in some it was not possible to give them such a defined appearance, this is because it was not possible to occupy vectorized images, so the quality of the images is somewhat low. , and therefore it is not possible to completely remove the background, nor have defined lines.

Code documentation

- **Loading models**

In order to visualize each of the modeled objects, it was necessary to load them in OpenGL.

```
//Carga de modelos
Model casa((char*)"Models/Casa/casaChicas.obj");
Model telefono((char*)"Models/Telefono/Phone2.obj");
Model telefonoArriba((char*)"Models/Telefono/Phone1.obj");
Model mesita((char*)"Models/Mesita/mesitaConTextura.obj");
Model cama((char*)"Models/Cama/Bed3.obj");
Model alfombra((char*)"Models/Alfombra2/alfombra2.obj");
Model lampara((char*)"Models/lampara2/lampara2.obj");
Model repisa((char*)"Models/Repisa/repisaLibros.obj");
Model puertaCasa((char*)"Models/Casa/puerta_casa.obj");
Model puertaCuarto((char*)"Models/Casa/puerta_cuarto.obj");
Model ventana((char*)"Models/Casa/ventana.obj");
Model ventanasCirculares((char*)"Models/Casa/ventana cirular.obj");
Model ventanaConBarrotes((char*)"Models/Casa/ventana con barrotes.obj");
Model barrotes((char*)"Models/Casa/barrotes.obj");
Model mueble((char*)"Models/Mueble/mueble.obj");
Model cajon1((char*)"Models/Mueble/cajon1.obj");
Model cajon2((char*)"Models/Mueble/cajon2.obj");
Model pelota((char*)"Models/Pelota/pelota.obj");
Model oso((char*)"Models/Oso de peluche/oso.obj");
```

- **Lights of developed ambient**

The characteristics of the light that was going to illuminate the environment were defined.

```
// Directional light
glUniform3f(glGetUniformLocation(lightingShader.Program, "dirLight.direction"), -0.2f, -1.0f, -0.3f);
glUniform3f(glGetUniformLocation(lightingShader.Program, "dirLight.ambient"), 0.8f, 0.8f, 0.8f);
glUniform3f(glGetUniformLocation(lightingShader.Program, "dirLight.diffuse"), 1.0f, 1.0f, 1.0f);
glUniform3f(glGetUniformLocation(lightingShader.Program, "dirLight.specular"), 1.0f, 1.0f, 1.0f);
```

- **Objects are drawn in the development environment**

This section shows how the different objects are commanded to draw, where there are some cases in which the basic transformations were required. Some objects had to be scaled to match the size of the house and other objects, most had to be translated to position them where they belonged, and others had to be rotated so they were aimed at a certain location. part of the house.

In the following image you can see the characteristics of some of the objects, among which are the house, the telephone and the top of the telephone.

The house was the most important object, since it was the base to be able to position all the other objects, therefore no type of transformation was applied to it.

The phone had to be separated into two parts, as it was one of the objects that would be given some animation and to move the speaker part it had to be kept

separate. Despite that, both parts were given the same scale and translation transformations.

A rotation has also been applied to the top, which will move the speaker as if the phone were receiving a call.

It is also important to mention that since the phone has eyes and a mouth, images were required from which the background had to be removed, so when drawing it, transparency is activated.

```
//Objetos
//Casa
model = glm::mat4(1);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
casa.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Telefono
model = glm::mat4(1);
model = glm::scale(model, glm::vec3(0.2f));
model = glm::translate(model, glm::vec3(39.5f, 87.0f, 51.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 1);
telefono.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);

//Telefono Arriba
model = glm::mat4(1);
model = glm::scale(model, glm::vec3(0.2f));
model = glm::translate(model, glm::vec3(39.5f, 87.0f, 51.0f));
model = glm::rotate(model, glm::radians(rotTelefonoArriba), glm::vec3(0.0f, 0.0f, 1.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
telefonoArriba.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
```

The next items shown are the coffee table with the phone on it, the bed, and the rug.

For the small table, only one translation had to be applied, since the size it had was adequate.

In the case of the bed, it was scaled, moved and turned to position it in its corresponding place.

The carpet was only scaled, but to be able to draw it it was also necessary to activate the transparency, since the background of the image that was used for its creation had to be eliminated, although it should be noted that there were some problems with the definition of the image. image, so the part of the background that is glued to the carpet was left unremoved and can be noticed in moments.

```
//Mesita
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(8.0f, 15.15f, 10.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
mesita.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Cama
model = glm::mat4(1);
model = glm::scale(model, glm::vec3(1.2f, 1.4f, 1.0f));
model = glm::translate(model, glm::vec3(0.0f, 10.5f, 5.3f));
model = glm::rotate(model, glm::radians(-90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
cama.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Alfombra
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(0.0f, 13.5f, -3.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 1);
alfombra.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
```

In the following image we have the corresponding for the lamp, the shelf and the door of the entrance of the house.

For the lamp it was also necessary to activate the transparency, but the same thing happened with the carpet, since the background could not be completely eliminated. Translation and rotation were applied to this object.

All three transformations were applied to the shelf with books. Whereas, the gate was only scaled and moved.

```
//Lampara
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(0.0f, 23.55f, 0.0f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 1);
lampara.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);

//Repisa
model = glm::mat4(1);
model = glm::scale(model, glm::vec3(1.8f, 1.8f, 0.7f));
model = glm::translate(model, glm::vec3(4.0f, 11.3f, 16.2f));
model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
repisa.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Puerta entrada casa
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(10.82f, 5.33f, 7.69f));
model = glm::rotate(model, glm::radians(rotPuertaFachada), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
puertaCasa.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
```

The following models that are available are the door of the room, the bars of the window on the left side of the house, the furniture in the room and one of the drawers

of the furniture. In general, some of the transformations were applied to all of them and it was not required to activate transparency.

```
//Puerta cuarto
model = glm::mat4(1);
model = glm::scale(model, glm::vec3(1.0f, 1.01f, 1.0f));
model = glm::translate(model, glm::vec3(-11.2f, 18.53f, -9.78f));
model = glm::rotate(model, glm::radians(rotPuertaCuarto), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
puertaCuarto.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Barrotes
model = glm::mat4(1);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
barrotes.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Mueble
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-0.75f, 14.39f, 5.2f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
mueble.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Cajón1 derecha
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(posicionCajon1, 14.39f, 3.5f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
cajon1.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
```

In this section there is the other drawer and a teddy bear, in both cases the same thing happens as in the previous objects.

```
//Cajón2 izquierda
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-6.45f, 14.39f, posicionCajon2));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
cajon2.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Oso
model = glm::mat4(1);
model = glm::scale(model, glm::vec3(0.8f));
model = glm::translate(model, glm::vec3(-12.8f, movOsoY, 1.9f));
model = glm::rotate(model, glm::radians(rotOso), glm::vec3(1.0f, 0.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
oso.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
```

After these objects, there are those in which the alpha channel had to be activated, since it was required that they have a certain translucency, since they were the windows that the house has.

```
glEnable(GL_BLEND);//Activa la funcionalidad para trabajar el canal alfa
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);

//Ventana fachada
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(10.7f, 9.1f, -1.65f));
model = glm::rotate(model, glm::radians(rot), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
glUniform4f(glGetUniformLocation(lightingShader.Program, "colorAlpha"), 1.0, 1.0, 1.0, 0.85);
ventana.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Ventanas circulares
model = glm::mat4(1);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
glUniform4f(glGetUniformLocation(lightingShader.Program, "colorAlpha"), 1.0, 1.0, 1.0, 0.85);
ventanasCirculares.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//Ventana con barrotes
model = glm::mat4(1);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
glUniform4f(glGetUniformLocation(lightingShader.Program, "colorAlpha"), 1.0, 1.0, 1.0, 0.85);
ventanaConBarrotes.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
```

Another object to which translucency was added was the ball, which received the three basic transformations.

```
//Pelota
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(movPelotaX, movPelotaY, movPelotaZ));
model = glm::rotate(model, glm::radians(45.0f), glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::rotate(model, glm::radians(rotPelota), glm::vec3(0.0f, 0.0f, 1.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glUniform1i(glGetUniformLocation(lightingShader.Program, "activaTransparencia"), 0);
glUniform4f(glGetUniformLocation(lightingShader.Program, "colorAlpha"), 1.0, 1.0, 1.0, 0.8);
pelota.Draw(lightingShader);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));


glDisable(GL_BLEND);  //Desactiva el canal alfa
glUniform4f(glGetUniformLocation(lightingShader.Program, "colorAlpha"), 1.0, 1.0, 1.0, 1.0);
glBindVertexArray(0);
```

After putting the characteristics of all the objects, the objects that have animations with the different keys were related, this to indicate when they wanted to start or end their movement. Here is an example of some:

```
//Para animación de objetos
if (keys[GLFW_KEY_R]) {
    animVentana1 = true;
}

if (keys[GLFW_KEY_F]) {
    animVentana2 = true;
}

if (keys[GLFW_KEY_T]) {
    animPuertaFachada1 = true;
}

if (keys[GLFW_KEY_G]) {
    animPuertaFachada2 = true;
}

if (keys[GLFW_KEY_Y]) {
    animPuertaCuarto1 = true;
}
```

That same function is used to give movement to some objects in which simple animations are applied, in these animations one or another transformation of the basic ones is performed.

```
if (keys[GLFW_KEY_R])
{
    if (animVentana1) {
        while (rot < 90) {
            rot += 0.1;
        }

    }
}

if (keys[GLFW_KEY_F])
{
    if (animVentana2) {
        while (rot > 0) {
            rot -= 0.1;
        }
    }
}

if (keys[GLFW_KEY_T])
{
    if (animPuertaFachada1) {
        while (rotPuertaFachada < 90) {
            rotPuertaFachada += 0.1;
        }

    }
}
```

Then you have another function called animation() and inside of which you have the complex animations.

These animations are complex because they were programmed as a state machine, both for the ball and for the phone, several cases were made to determine the position of the object in a state.

```
void animacion(){
    if (pelotaEnMovimiento)
    {
        if (movPelota1)
        {
            movPelotaY += 0.1f;
            if (movPelotaY > 17.4f) {
                movPelota1 = false;
                movPelota2 = true;
            }
        }
        if (movPelota2)
        {
            movPelotaY -= 0.1f;
            movPelotaX -= 0.05f;
            if (movPelotaY < 14.5f && movPelotaX < 4.5f)
            {
                movPelota2 = false;
                movPelota3 = true;
            }
        }
        if (movPelota3)
        {
            movPelotaY += 0.2f;
            movPelotaX -= 0.1f;
            if (movPelotaY > 16.0f && movPelotaX < 3.0f)
            {
                movPelota3 = false;
                movPelota4 = true;
            }
        }
```

Conclusions

With the realization of this project it was possible to understand in greater detail how graphic computing works, what aspects influence the development of graphic environments and what is the importance of knowing how to use the different tools that we have at our disposal to achieve greater results in carrying out projects of this type. It seems to me that it is an area of computing that covers a lot of knowledge, but the results of the development of environments are quite satisfactory. Personally, I really liked doing this recreation of the house, although I have to accept that it is an area that requires a lot of work, study, concentration, but above all time.