

PROYECTO FINAL

CASA DE JOHNNY BRAVO



6 DE NOVIEMBRE DE 2019

LABORATORIO DE COMPUTACIÓN GRÁFICA E INTERACCIÓN HUMANO-COMPUTADORA
FABIAN CASTILLO FANNY ANAHI

Para el presente proyecto el objetivo fue realizar el cuarto mostrado al profesor, en este caso, elegí la siguiente.

Por fuera:



Por dentro:



El entorno, así como la casa y algunos detalles como algunas puertas y ventanas fueron realizadas con primitivas. En total utilice 14 texturas diferentes.

```
// Load textures
GLuint texture1, texture2, texture3, texture4, texture5, texture6, texture7, texture8, texture9, texture10,
        texture11, texture12, texture13, texture14;
glGenTextures(1, &texture1);
glGenTextures(1, &texture2);
glGenTextures(1, &texture3);
glGenTextures(1, &texture4);
glGenTextures(1, &texture5);
glGenTextures(1, &texture6);
glGenTextures(1, &texture7);
glGenTextures(1, &texture8);
glGenTextures(1, &texture9);
glGenTextures(1, &texture10);
glGenTextures(1, &texture11);
glGenTextures(1, &texture12);
glGenTextures(1, &texture13);
glGenTextures(1, &texture14);
```

```

318 // TEXTURA 1
319 image = stbi_load("images/VERDE.jpg", &texturewidth, &textureheight, &nChannels, 0);
320 glBindTexture(GL_TEXTURE_2D, texture);
321 glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, texturewidth, textureheight, 0, GL_RGB, GL_UNSIGNED_BYTE, image);
322 glGenerateMipmap(GL_TEXTURE_2D);
323 if (image)
324 {
325     glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, texturewidth, textureheight, 0, GL_RGB, GL_UNSIGNED_BYTE, image);
326     glGenerateMipmap(GL_TEXTURE_2D);
327 }
328 else
329 {
330     std::cout << "Failed to load texture" << std::endl;
331 }
332 stbi_image_free(image);
333
334 // TEXTURA 2
335 glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
336 glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
337 glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
338 glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST_MIPMAP_NEAREST);
339 image = stbi_load("images/cereal.jpg", &texturewidth, &textureheight, &nChannels, 0);
340 glBindTexture(GL_TEXTURE_2D, texture2);
341 glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, texturewidth, textureheight, 0, GL_RGB, GL_UNSIGNED_BYTE, image);
342 glGenerateMipmap(GL_TEXTURE_2D);
343 if (image)

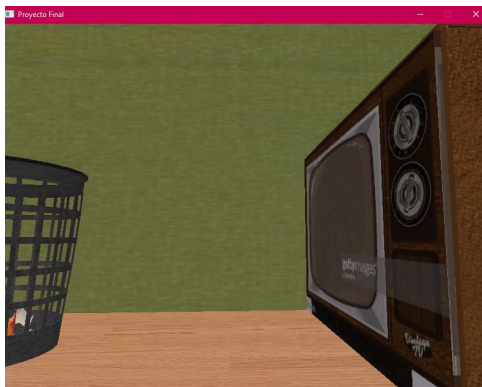
```

Debido a que al menos un objeto tenía que ser elaborado el OpenGL, yo decidí realizar la televisión, para la cual utilicé dos texturas diferentes.

```

//////////////////////////////////// TV frontal
glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_2D, texture14);
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(28.71f, 2.98f, -14.3f)); //28.71f, 3.48f, -14.1f
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 0.0f, 1.0f));
model = glm::scale(model, glm::vec3(0.2f, 7.0f, 5.0f)); //0.2f, 7.0f, 6.0f
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glDrawArrays(GL_TRIANGLES, 0, 36);
//////////////////////////////////// TV superior
glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_2D, texture11);
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(31.71f, 5.5f, -14.3f));
model = glm::scale(model, glm::vec3(6.0f, 0.2f, 7.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glDrawArrays(GL_TRIANGLES, 0, 36);
//////////////////////////////////// TV trasera
glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_2D, texture14);
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(34.7f, 2.98f, -14.3f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(1.0f, 0.0f, 0.0f));
model = glm::scale(model, glm::vec3(0.2f, 7.0f, 5.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glDrawArrays(GL_TRIANGLES, 0, 36);
//////////////////////////////////// Tvlateral no visible
glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_2D, texture11); //cambia textura a cafe
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(31.71f, 2.98f, -17.7f));
model = glm::scale(model, glm::vec3(6.0f, 5.0f, 0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glDrawArrays(GL_TRIANGLES, 0, 36);
//////////////////////////////////// Tvlateral visible
glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_2D, texture11); //cambia textura a cafe
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(31.71f, 2.98f, -10.8f));
model = glm::scale(model, glm::vec3(6.0f, 5.0f, 0.2f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glDrawArrays(GL_TRIANGLES, 0, 36);

```



Para continuar con el proyecto, importe en total 13 objetos.

```

//////////////////////////////////// LOAD MODELS //////////////////////////////////////
Model ourModel1((char *) "Models/modeloproyecto/ventanaeneltecho.obj"); //LISTO
Model ourModel12((char *) "Models/modeloproyecto/ventanaTechoInterior.obj"); //LISTO
Model ourModel14((char *) "Models/modeloproyecto/arbolfEscalado.obj"); //LISTO
Model ourModel13((char *) "Models/modeloproyecto/silla.obj"); //LISTO
Model ourModel15((char *) "Models/modeloproyecto/cajonera8.obj"); //LISTO
Model ourModel16((char *) "Models/modeloproyecto/mesaRojoOxido.obj"); //LISTO
Model ourModel17((char *) "Models/modeloproyecto/lampara1.obj"); //LISTO
Model ourModel18((char *) "Models/modeloproyecto/camaLmoHada.obj"); //LISTO
Model ourModel19((char *) "Models/modeloproyecto/basura.obj"); //LISTO

Model ourModel10((char *) "Models/modeloproyecto/nubes.obj"); // NUBES
Model ourModel11((char *) "Models/modeloproyecto/bird.obj"); // PERRO

Model ourModel12((char *) "Models/modeloproyecto/windowCirc.obj");

Model ourModel17((char *) "Models/modeloproyecto/dog.obj"); // PERRO

```

De los cuales tres fueron animados fuera de la casa: nubes, pajar y perro.

```

void animacion1() //PAJARO
{
    if (circuito)
    {
        if (recorrido1)
        {
            rotKit3 = 0.0;
            movKitX3 += 0.1f;
            movKitZ3 -= 0.1f; //46.0f, 0.0f, -23.0f
            if (movKitX3 > 26 && movKitZ3 < -13)
            {
                recorrido1 = false;
                recorrido2 = true;
            }
        }
        if (recorrido2)
        {
            rotKit3 = 270.0; //46.0f, 0.0f, -23.0f
            movKitX3 -= 0.1f;
            if (movKitX3 < -51)
            {
                recorrido2 = false;
                recorrido1 = true;
            }
        }
    }
}

void animacion2() //NUBES
{
    if (circuito)
    {
        if (recorrido3)
        {
            movKitZ2 -= 0.05f;
            if (movKitZ2 < -10.0)
            {
                recorrido3 = false;
                recorrido4 = true;
            }
        }
        if (recorrido4)
        {
            movKitZ2 += 0.05f;
            if (movKitZ2 > 31)
            {
                recorrido3 = true;
                recorrido4 = false;
            }
        }
    }
}

animacion3() //Perro
{
    if (circuito)
    {
        if (recorrido5)
        {
            //rotKit = 180.0;
            movKitZ = 0.1f;
            if (movKitZ < -33)
            {
                recorrido5 = false;
                recorrido6 = true;
            }
        }
        if (recorrido6)
        {
            rotKit = 90.0;
            movKitX += 0.1f;
            if (movKitX > 33.0)
            {
                recorrido6 = false;
                recorrido7 = true;
            }
        }
        if (recorrido7)
        {
            rotKit = 0.0; //////////////////////////////////////////////////
            movKitZ += 0.1f;
        }
    }
}

```

Para iniciar las animaciones, es necesario presionar la tecla 'U'.



- Las nubes se mueven constantemente hacia atrás y hacia enfrente.
- El pajarito hace un recorrido desde su posición inicial, volando en dirección hacia el árbol derecho y finalmente termina posándose sobre el árbol izquierdo.
- El recorrido del perro es el más largo de los tres debido a que avanza desde su posición inicial, camina hacia el jardín y le da una vuelta a la casa hasta detenerse frente a la puerta.

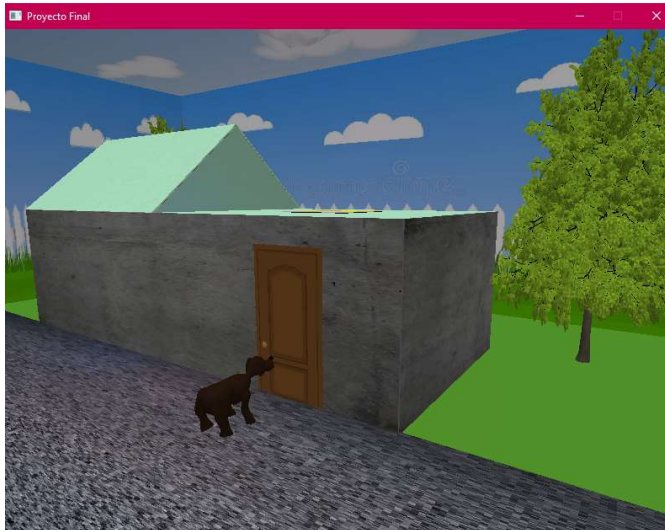
Posición inicial de los tres objetos:



Movimiento de nubes y pajarito:



Fin del recorrido del perro:



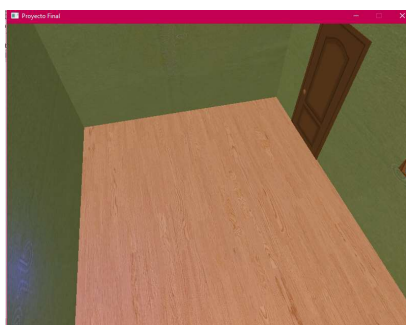
Resultado final del interior del cuarto.





En el cuarto amueblado (cuarto derecho) entra una luz desde la venta circular y refleja en la pared que tiene enfrente, igualmente con la ventana trasera del cuarto no amueblado (cuarto izquierdo) hay una luz como si entrara desde la ventana.

Cuarto izquierdo:



Cuarto derecho:



Interaccion dentro del ambiente

TECLA	FUNCIÓN
W / ARRIBA S / ABAJO A / IZQUIERDA D / DERECHA	CONTROL DE CÁMARA
U	ANIMACIÓN DE LOS TRES OJETOS (no es necesario mantener presionada).
MOUSE TOUCHPAD	DIRIGIR EL ENFOQUE DE LA CAMARA

Link para descargar el proyecto

<https://github.com/7FannyAnahi/ProyectoGrafica>

Conclusiones:

Con este proyecto tube la oportunidad de practicar un poco más la manipulación de los planos en OpenGL, al igual que las animaciones, sobre todo porque a diferencia de las practicas realizadas durante el semestre las posiciones iniciales ya estaban dadas.

Otro punto que inicialmente me costo un poco de trajo fue acomodar la posición inicial de la cámara para enfocar justo el punto en donde yo quería iniciar, pues anteriormente estaba como la utilizada en la practica en donde animamos el lamorghini.

Otro aspecto que alentaba el proceso para la elaboración del proyecto fue la búsqueda de texturas que sirvieran en OpenGL, muchas de ellas eran inserbibles.

Finalmente, aquí podemos observar el trabajo final, que en lo personal, fue muy satisfactorio.