UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e INTERACCIÓN HUMANO COMPUTADORA

Documentación técnica proyecto final : Campo de comida Nacional e Internacional

Nombre completo:

Salazar Serrano Edgar

N° de cuenta: 416101630

Grupo de Laboratorio: 10

Grupo de teoría: 6

Semestre 2023-1

Objetivo

Utilizar las herramientas y los conocimientos adquiridos en el laboratorio de computación Gráfica e interacción humano computadora para recrear un escenario virtual usando el lenguaje de programación C++ y Open GL, esto con la temática de puestos de comida nacionales, en donde el usuario pueda recorrerlo libremente e interactuar con un avatar y algunos objetos que tienen una función especifica

Alcance

Se pretende recrear un espacio 3d que simule una feria virtual que incluye puestos de comida, personajes, luces y animaciones, interacción con el usuario por medio de teclas y adicionalmente agregamos efectos de sonido.

Descripción técnica

1. Creación de objetos y texturizado

Para la creación de nuestros objetos en Open GL fue posible hacerlo mediante carga de modelos prediseñados en software de modelado 3D y diseñados por nosotros con primitivas geométricas de forma jerárquica en OpenGL .

Para texturizar los modelos se hace mediante imágenes que se cargan por separado a los modelos y se declaran dentro del código, en el caso de los modelos ya hechos en software 3d, se tienen sus respectivas texturas y solo hay que cargarlos.

Declaración de modelos y texturas.

```
Model jimmy_cabeza;
Model jimmy_tronco;
Model pierna_izq;
Model pierna_izq_abajo;
Model pierna_der;
Model pierna_der_abajo;
Model brazo_izq;
Model brazo_der;
Model mano_der;
Model mano_izq;
```

```
Texture mariscos;
Texture puesto;
Texture mariscos_comida;
Texture helados_caja;
Texture tortas_rotulo;
Texture tortas_front;
Texture rotulo_represion;
Texture tacos_rotulo;
Texture tacos_front;
Texture tacos_rotulo_mini;
Texture jugos_rotulo;
Texture jugos_front;
Texture jugos_mini;
Texture fuego;
```

Se declaran tambien la ubicación de los modelos y las texturas

ejemplo:

```
jugos_rotulo = Texture("Textures/jugos_rotulo.png");
jugos_rotulo.LoadTextureA();
jugos_front = Texture("Textures/jugos_front.png");
jugos_front.LoadTextureA();
jugos_mini = Texture("Textures/jugos_rotulo_mino.png");
jugos_mini.LoadTextureA();
```

```
jimmy = Model();
jimmy.LoadModel("Models/jimmy.obj");

aku = Model();
aku.LoadModel("Models/Aku_Aku_READY.fbx");

shrek = Model();
shrek.LoadModel("Models/CHARACTER_Shrek.obj");
```

modelos a recrear:

1- puestos callejeros de comida

- puesto de tortas
- puesto de tacos
- puesto de jugos
- puestos mariscos
- camión de helados
- cohete espacial

imagenes de referencia para los puestos y el camión de helados:

2- personajes
    - shrek
    - Jimmy Neutron (Avatar)

Imágenes de Referencia

Para crear los puestos los diseñamos con figuras geométricas hechas en OpenGL principalmente cubos, en los cuales se les modificó la escala para obtener rectángulos de diversos tamaños, asi creamos los puestos y el camión de helados de manera jerárquica para poder mover, rotar o escalar los objetos libremente sin que se pierda la forma original.

Definición de un cubo en open GL:

```
unsigned int cubo_indices[] = {
    // front
    0, 1, 2,
    2, 3, 0,
    // right
    4, 5, 6,
    6, 7, 4,
    // back
    8, 9, 10,
    10, 11, 8,

    // left
    12, 13, 14,
    14, 15, 12,
    // bottom
    16, 17, 18,
    18, 19, 16,
    // top
    20, 21, 22,
    22, 23, 20,
};
```

```
GLfloat cubo_vertices[] = {
    //x     y       z       S       T           NX      NY      NZ
    -0.5f, -0.5f,  0.5f,    0.0f,  0.0f,       0.0f,   0.0f,   -1.0f,  //0
    0.5f, -0.5f,  0.5f,     1.0f,   0.0f,      0.0f,   0.0f,   -1.0f,  //1
    0.5f,  0.5f,  0.5f,     1.0f,   1.0f,      0.0f,   0.0f,   -1.0f,  //2
    -0.5f,  0.5f,  0.5f,    0.0f,   1.0f,      0.0f,   0.0f,   -1.0f,  //3
    // right
    //x     y       z       S       T
    0.5f, -0.5f,  0.5f,     0.0f,  0.0f,       -1.0f,  0.0f,   0.0f,
    0.5f, -0.5f,  -0.5f,    1.0f,   0.0f,      -1.0f,  0.0f,   0.0f,
    0.5f,  0.5f,  -0.5f,    1.0f,   1.0f,      -1.0f,  0.0f,   0.0f,
    0.5f,  0.5f,  0.5f,     0.0f,   1.0f,      -1.0f,  0.0f,   0.0f,
    // back
    -0.5f, -0.5f, -0.5f,    0.0f,   0.0f,      0.0f,   0.0f,   1.0f,
    0.5f, -0.5f, -0.5f,     1.0f,   0.0f,      0.0f,   0.0f,   1.0f,
    0.5f,  0.5f, -0.5f,     1.0f,   1.0f,      0.0f,   0.0f,   1.0f,
    -0.5f,  0.5f, -0.5f,    0.0f,   1.0f,      0.0f,   0.0f,   1.0f,

    // left
    //x     y       z       S       T
    -0.5f, -0.5f,  -0.5f,   0.0f,  0.0f,       1.0f,   0.0f,   0.0f,
    -0.5f, -0.5f,  0.5f,    1.0f,   0.0f,      1.0f,   0.0f,   0.0f,
    -0.5f,  0.5f,  0.5f,    1.0f,   1.0f,      1.0f,   0.0f,   0.0f,
    -0.5f,  0.5f,  -0.5f,   0.0f,   1.0f,      1.0f,   0.0f,   0.0f,

    // bottom
    //x     y       z       S       T
    -0.5f, -0.5f,  0.5f,    0.0f,  0.0f,       0.0f,   1.0f,   0.0f,
    0.5f,  -0.5f,  0.5f,    1.0f,   0.0f,      0.0f,   1.0f,   0.0f,
     0.5f,  -0.5f,  -0.5f,  1.0f,   1.0f,      0.0f,   1.0f,   0.0f,
    -0.5f, -0.5f,  -0.5f,   0.0f,   1.0f,      0.0f,   1.0f,   0.0f,

    //UP
    //x     y       z       S       T
     -0.5f, 0.5f,  0.5f,    0.0f,  0.0f,       0.0f,   -1.0f,  0.0f,
      0.5f,  0.5f,  0.5f,   1.0f,   0.0f,      0.0f,   -1.0f,  0.0f,
       0.5f, 0.5f,  -0.5f,  1.0f,   1.0f,      0.0f,   -1.0f,  0.0f,
      -0.5f, 0.5f,  -0.5f,  0.0f,   1.0f,      0.0f,   -1.0f,  0.0f,
};
```

```
Mesh* obj6 = new Mesh();
obj6->CreateMesh(cubo_vertices, cubo_indices, 256, 96);
meshList.push_back(obj6);
```

Despues para instanciar los objetos se declara una matriz para el modelo, la cual tendrá la información de las transformaciones geométricas que le hagamos, además de que se le asignan las texturas usadas en cada modelo:

Ejemplo para instanciar un objeto:

```
// puesto  tortas

    color = glm::vec3(1.0f, 1.0f, 1.0f);
    model = glm::mat4(1.0);
    model = glm::translate(model, glm::vec3(-60.0f, 1.0f, -10.0f));
    modelaux = model;
    model = glm::scale(model, glm::vec3(5.0f, 2.0f, 3.0f) * plus);
    glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
    glUniform3fv(uniformColor, 1, glm::value_ptr(color));
    puesto.UseTexture();
    Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
    meshList[5]->RenderMesh();
    // parte superior
    color = glm::vec3(1.0f, 1.0f, 1.0f);
    model = modelaux;
    model = glm::translate(model, glm::vec3(0.0f, 2.0f, 0.0f) * plus);
    model = glm::scale(model, glm::vec3(5.0f, 2.0f, 3.0f) * plus);
    glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
    glUniform3fv(uniformColor, 1, glm::value_ptr(color));
    tortas_rotulo.UseTexture();
    Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
    meshList[5]->RenderMesh();
```

Siguiendo esta técnica tenemos objetos 3d creados por modelado geométrico y jerárquico

2. Iluminación

Para lograr que las formas creadas puedan verse y tengan efecto de profundidad se hace uso de la iluminación. Se tienen 3 tipos de iluminación: Point Light, Spot Light y Dirección light.

En este escenarios usamos una luz direccional (Direction Ligth) que funciona como luz ambiental por ejemplo, la luz del sol. Tambien usamos luces tipo SpotLight que funcionan como las luces de un faro.

Declaración de Direction light:

Como parámetro se definió una luz blanca (codigo RGB = (1,1,1)) , la luz tiene una dirección negativa en el eje Y y negativa en el eje Z.

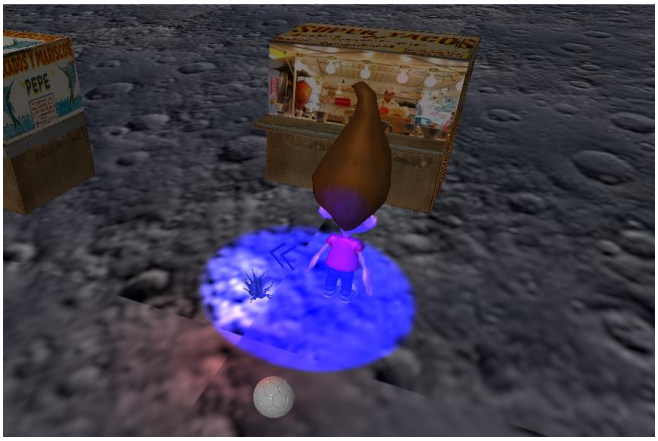
```
mainLight = DirectionalLight(1.0f, 1.0f, 1.0f,
    0.3f, 0.3f,
    0.0f, -1.0f, -1.0f);
```

Declaración de Spotlight:

Se modifican los parámetros de color, dirección posición para la iluminación

```
spotLights[2] = SpotLight(1.0f, 1.0f, 1.0f,
    0.0f, 2.0f,
    0.0f, 0.0f, 0.0f,
    0.0f, -5.0f, 0.0f,
    1.0f, 0.0f, 0.0f,
    50.0f);
spotLightCount++;
```

Resultados de iluminación



Bibliografía

- Times, T. N. Y. (2022, 30 agosto). *El efímero arte del rótulo en Ciudad de México*. The New York Times. Recuperado 5 de octubre de 2022, de https://www.nytimes.com/es/interactive/2022/07/21/espanol/ciudad-de-mexico-rotulos.html