UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e INTERACCIÓN HUMANO COMPUTADORA

EJERCICIOS DE CLASE Nº 08

NOMBRE COMPLETO: Reyes Romero Luis Fernando

Nº de Cuenta: 318155320

GRUPO DE LABORATORIO: 11

GRUPO DE TEORÍA: 06

SEMESTRE 2024-2

FECHA DE ENTREGA LÍMITE: 06 de abril de 2024

CALIFICACIÓN: _____

EJERCICIOS DE SESIÓN:

Ejercicios

- Agregar su dado de 10 caras y editar sus normales para que las caras del dado sean iluminadas correctamente.
- Apagar con teclado la luz (pointlight) de su lámpara creada para el reporte de la práctica 7.

Para el ejercicio del dado lo primero que hacemos es copiar el mismo dado con sus coordenadas de textura que se hizo en la práctica 6 de texturizado. Una vez hecho eso, vamos a modificar sus normales para que la luz que esta ligada a la cámara pueda iluminar de forma correcta al dado. Dicho esto, debemos tener en cuenta de que lado o como estamos viendo cada cara del dado para tener en cuenta la normal que se requiera iluminar. A continuación, se muestran las modificaciones necesarias para la iluminación del dado. Cabe mencionar, que para las normales en el eje y para las caras superiores del dado son -1.0f, ya que necesitamos ver la iluminación del dado desde arriba apuntando desde abajo. En cambio, en las caras inferiores se debe de iluminar en 1.0f en el eje y, ya que ahora necesitamos iluminar el dado desde arriba.

```
void CrearDado10()
{
    unsigned int Dado10_indices[] = {
        //top
        0,  1,  2,
        3,  4,  5,
        6,  7,  8,
        9, 10, 11,
       12, 13, 14,

       //bottom
       15, 16, 17,
       18, 19, 20,
       21, 22, 23,
       24, 25, 26,
       27, 28, 29,

    };
```

```
GLfloat Dado10_vertices[] = {
    // Parte de arriba
    //Cara 1
    //x     y       z       S       T          NX       NY       NZ
    -0.5f, -0.5f,  0.5f,    0.0f,    0.55f,     0.0f,   -1.0f,   -1.0f,
     0.5f, -0.5f,  0.5f,    0.2f,    0.55f,     0.0f,   -1.0f,   -1.0f,
     0.0f,  0.5f, -0.35f,   0.1f,    1.0f,      0.0f,   -1.0f,   -1.0f,

     //Cara 2
     0.5f, -0.5f,  0.5f,    0.38f,   0.55f,    -1.0f,   -1.0f,   -1.0f,
     1.0f, -0.5f, -0.5f,    0.6f,    0.55f,    -1.0f,   -1.0f,   -1.0f,
     0.0f,  0.5f, -0.35f,   0.50f,   1.0f,     -1.0f,   -1.0f,   -1.0f,

     //Cara 3
     1.0f, -0.5f, -0.5f,    0.6f,    0.0f,     -1.0f,   -1.0f,    1.0f,
     0.0f, -0.5f, -1.2f,    0.8f,    0.0f,     -1.0f,   -1.0f,    1.0f,
     0.0f,  0.5f, -0.35f,   0.7f,    0.5f,     -1.0f,   -1.0f,    1.0f,

     //Cara 4
     0.0f, -0.5f, -1.2f,    0.8f,    0.55f,     1.0f,   -1.0f,    1.0f,
    -1.0f, -0.5f, -0.5f,    1.00f,   0.55f,     1.0f,   -1.0f,    1.0f,
     0.0f,  0.5f, -0.35f,   0.9f,    1.00f,     1.0f,   -1.0f,    1.0f,

     //Cara 5
    -1.0f, -0.5f, -0.5f,    0.2f,    0.00f,     1.0f,   -1.0f,   -1.0f,
    -0.5f, -0.5f,  0.5f,    0.4f,    0.00f,     1.0f,   -1.0f,   -1.0f,
     0.0f,  0.5f, -0.35f,   0.3f,    0.50f,     1.0f,   -1.0f,   -1.0f,
```

```
    //Parte de Abajo
    //Cara 1
    //x     y       z       S       T          NX       NY       NZ
    -0.5f, -0.5f,  0.5f,    0.4f,    0.55f,     0.0f,    1.0f,   -1.0f,
     0.5f, -0.5f,  0.5f,    0.2f,    0.55f,     0.0f,    1.0f,   -1.0f,
     0.0f, -1.5f, -0.35f,   0.3f,    1.0f,      0.0f,    1.0f,   -1.0f,

    //Cara 2
     0.5f, -0.5f,  0.5f,    0.38f,   0.0f,     -1.0f,    1.0f,   -1.0f,
     1.0f, -0.5f, -0.5f,    0.6f,    0.0f,     -1.0f,    1.0f,   -1.0f,
     0.0f, -1.5f, -0.35f,   0.5f,    0.5f,     -1.0f,    1.0f,   -1.0f,

    //Cara 3
     1.0f, -0.5f, -0.5f,    1.0f,    0.0f,     -1.0f,    1.0f,    1.0f,
     0.0f, -0.5f, -1.2f,    0.8f,    0.0f,     -1.0f,    1.0f,    1.0f,
     0.0f, -1.5f, -0.35f,   0.9f,    0.6f,     -1.0f,    1.0f,    1.0f,

    //Cara 4
     0.0f, -0.5f, -1.2f,    0.80f,  0.55f,     1.0f,    1.0f,    1.0f,
    -1.0f, -0.5f, -0.5f,    0.60f,  0.55f,     1.0f,    1.0f,    1.0f,
     0.0f, -1.5f, -0.35f,   0.70f,  1.00f,     1.0f,    1.0f,    1.0f,

    //Cara 5
    -1.0f, -0.5f, -0.5f,    0.20f,  0.00f,     1.0f,    1.0f,   -1.0f,
    -0.5f, -0.5f,  0.5f,    0.00f,  0.00f,     1.0f,    1.0f,   -1.0f,
     0.0f, -1.5f, -0.35f,   0.10f,  0.55f,     1.0f,    1.0f,   -1.0f,
```

```
//Dado de 10 caras
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(-5.0f, 1.0f, -2.5f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Dado10Texture.UseTexture();
meshList[4]->RenderMesh();
```

Una vez hecho estos cambiamos compilamos el código y observamos como se está iluminando el dado.

Compilación del código de la iluminación del dado

De igual forma, en los entregables se va a adjuntar un vídeo donde se muestre de mejor forma la iluminación del dado.

Segundo ejercicio

Para el segundo ejercicio lo que se nos pide es que por medio del teclado seamos capaces de apagar la luz que le colocamos a nuestra lámpara. Lo primero que vamos a hacer es ir a nuestro archivo Window.h ahí vamos a definir dos funciones que nos van a ayudar a realizar la acción que nos pide.

```
// Función para obtener el estado de apagar (si es true o false)
bool GetApagarLuz() { return apagar; }
// Función para establecer el estado de apagar (true o false)
void ApagarLuz(bool state) { apagar = state; }
```

Como podemos observar se crea la función GetApagarLuz() de tipo booleana, ya que solo va a tomar el valor de verdadero o falso y regresa la variable apagar y debajo de esa función hay otra función que es ApagarLuz(bool state) donde esta función va a tomar como parámetro verdadero o falso y este valor se le asignará a la variable apagar. Cabe mencionar, que todo esto se hace en la parte pública del código, ya que estas funciones las necesitaremos llamar en los códigos Window.cpp y en el archivo main. Dicho esto, ahora vamos a inicializar nuestra variable apagar y esa la definimos en la parte privada del archivo Window.h y la definimos como “false”, ya que al compilar el código es falso que la luz este apagada por esa razón se inicializa apagar como false.

```
private:
    GLFWwindow *mainWindow;
    GLint width, height;
    bool keys[1024];
    GLint bufferWidth, bufferHeight;
    void createCallbacks();
    GLfloat lastX;
    GLfloat lastY;
    GLfloat xChange;
    GLfloat yChange;
    GLfloat muevex;
    GLfloat muevex2;
    bool apagar = false;
    bool mouseFirstMoved;
    static void ManejaTeclado(GLFWwindow* window, int key, int code, int action, int mode);
    static void ManejaMouse(GLFWwindow* window, double xPos, double yPos);

};
```

Después de hacer estos cambios en el archivo Window.h ahora nos vamos al archivo Window.cpp en especifico en la sección del manejo del teclado.

```
if (key == GLFW_KEY_Z)
{
    theWindow->ApagarLuz(true);
}
if (key == GLFW_KEY_X)
{
    theWindow->ApagarLuz(false);
}
```

Como podemos observar estamos diciendo que al seleccionar la tecla Z la luz de la lámpara se tendrá que apagar y si presionamos la tecla X la luz de la lámpara se de debe de prender. De igual forma, nuestra función que ponemos en la condición es la de ApagarLuz(bool state) y como ya habíamos dicho anteriormente el valor que le pasemos a esta función será el valor que tome la variable apagar esto es muy importante, ya que el valor de la variable apagar es la encargada de decidir si la función GetApagarLuz() se cumple o no. Después de hacer estos cambios nos vamos a nuestro archivo main y dentro de nuestro while en la sección donde se le pasa la información al shader de las fuentes de iluminación colocamos la siguiente condición.

```
//Condición para apagar la luz de nuestra lámpara
if (mainWindow.GetApagarLuz()) {
    shaderList[0].SetPointLights(pointLights, pointLightCount-1);
}
```

Esta condición lo que nos dice es que si se cumple GetApagarLuz() (Solo se puede o no cumplir esta función debido a que es una función booleana y solo toma valores de verdadero o falso) se cumple la condición y se ejecuta lo que esta dentro de nuestro if. Dentro del if la condición es que la información de las luces de tipo puntual se las pasa al shader excepto la información de la última luz de tipo puntual. Si nos vamos a la declaración de las luces puntuales podemos ver que la última luz de este tipo es la de la lámpara.

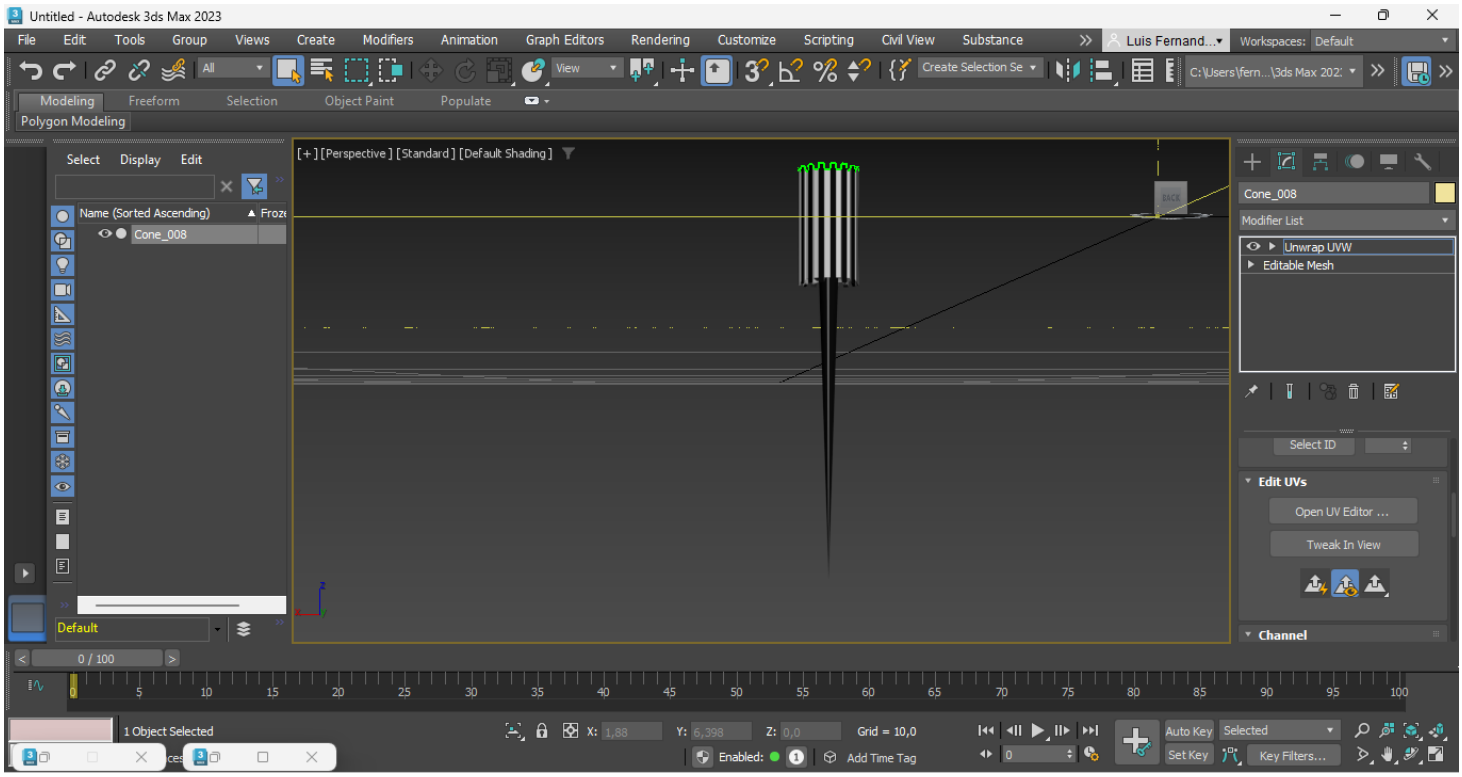
```
unsigned int pointLightCount = 0;
//Declaración de primer luz puntual
pointLights[0] = PointLight(1.0f, 0.0f, 0.0f,
    0.0f, 1.0f,
    -6.0f, 1.5f, 1.5f,
    0.3f, 0.2f, 0.1f);
pointLightCount++;

//luz de la lampara
pointLights[1] = PointLight(1.0f, 1.0f, 1.0f,
    0.3f, 0.3f,
   -1.0f, 0.0f, 12.0f,
    0.3f, 0.2f, 0.1f);
pointLightCount++;
```

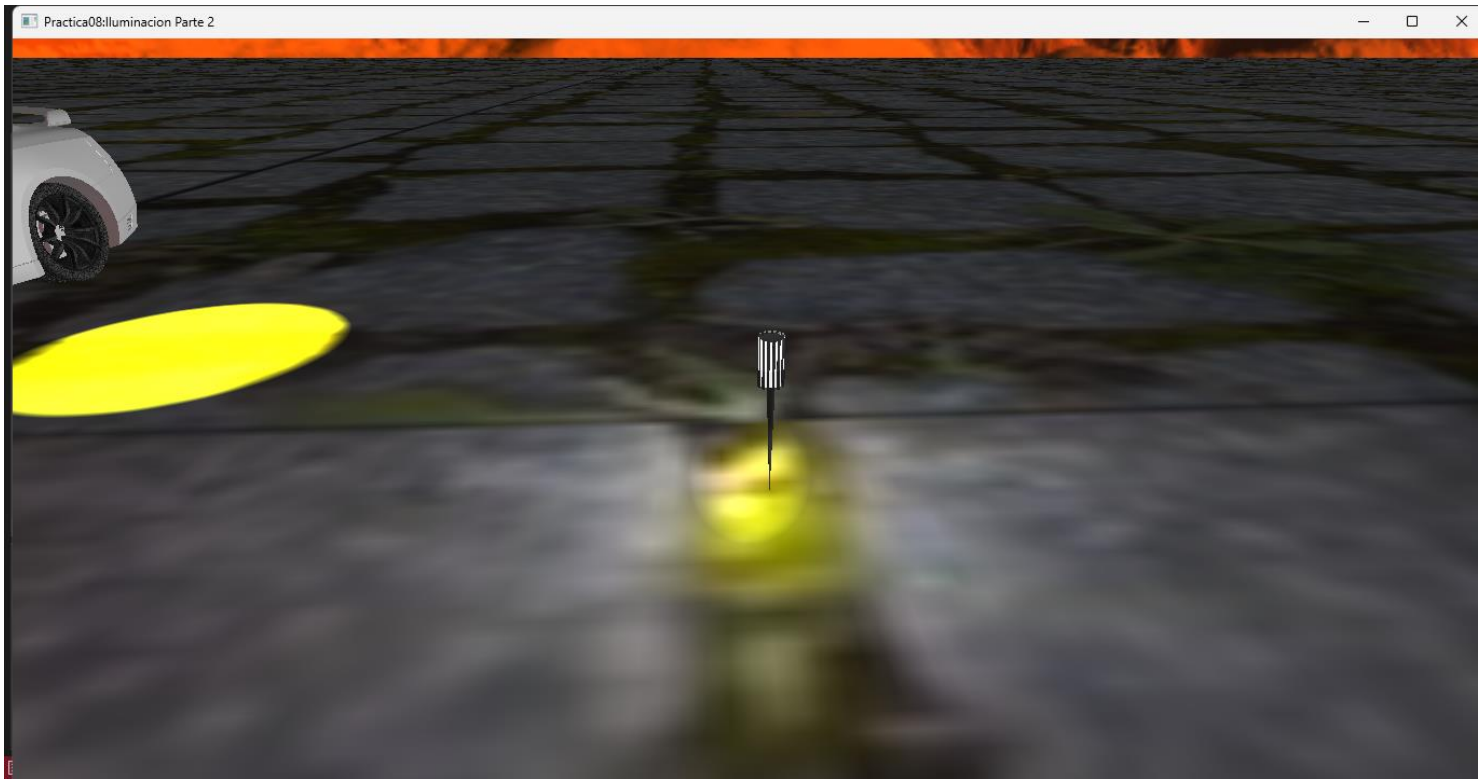
Entonces, lo que hace esta condición es que al cumplirse GetApagarLuz() la información de la luz de la lampara la ignora totalmente el shader por esa razón se da la ilusión de que se apaga la luz. Cabe mencionar, que en nuestro archivo Window.cpp colocamos que al apretar la tecla Z la función ApagarLuz(bool state) es verdadera esto quiere decir que se cumple la función GetApagarLuz() y se cumple la condición del if y se apaga la luz de la lámpara. En cambio, si apretamos la tecla X la función ApagarLuz(bool state) se vuelve false y la función GetApagarLuz() se vuelve falsa, por ende, no toma en cuenta la condición del if, ya que GetApagarLuz() es falsa y la luz de la lámpara se enciende.

Capturas de pantalla

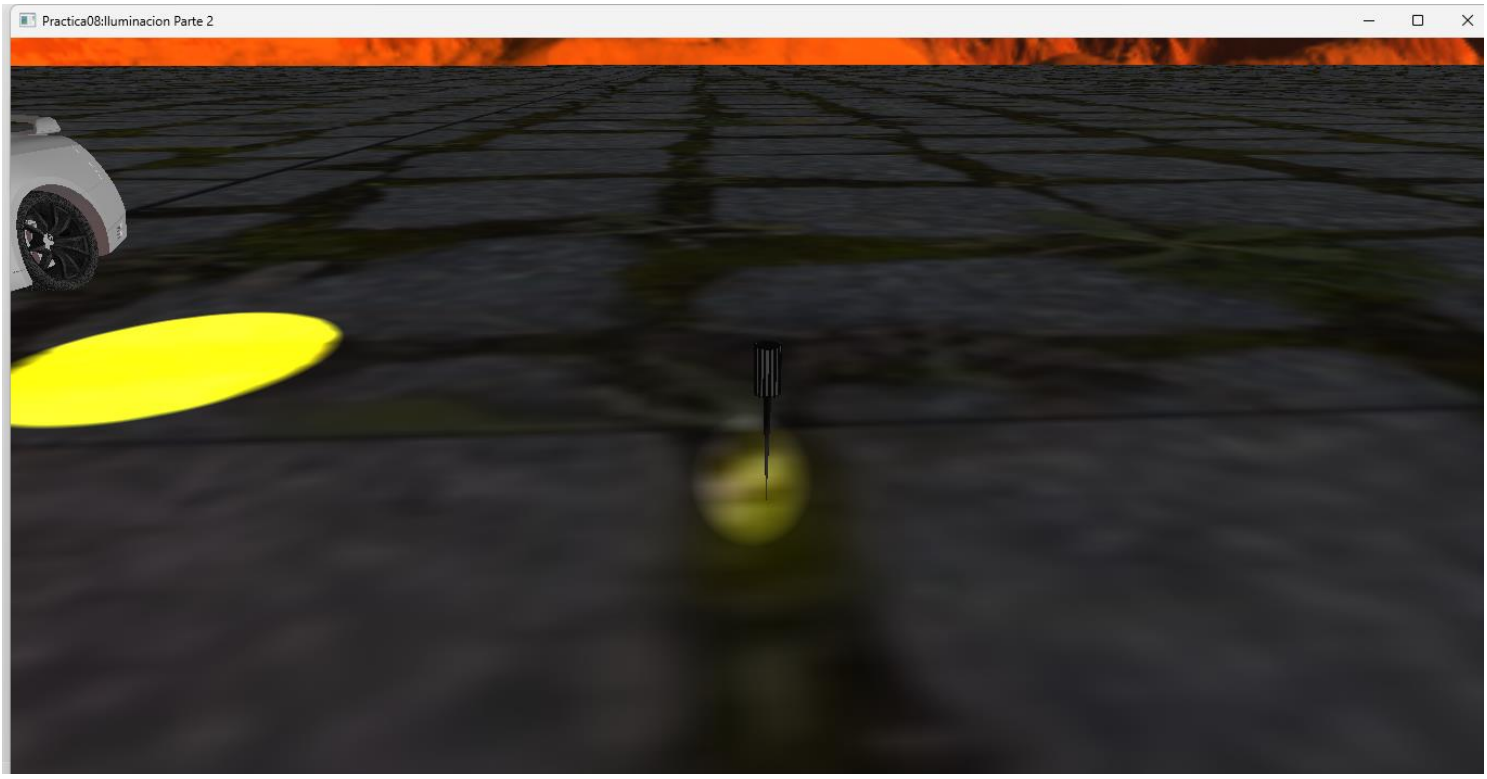
Un dato importante es que en la práctica anterior no texturizamos nuestra lámpara, pero ahora si la texturizamos con ayuda de nuestro software 3ds Max.



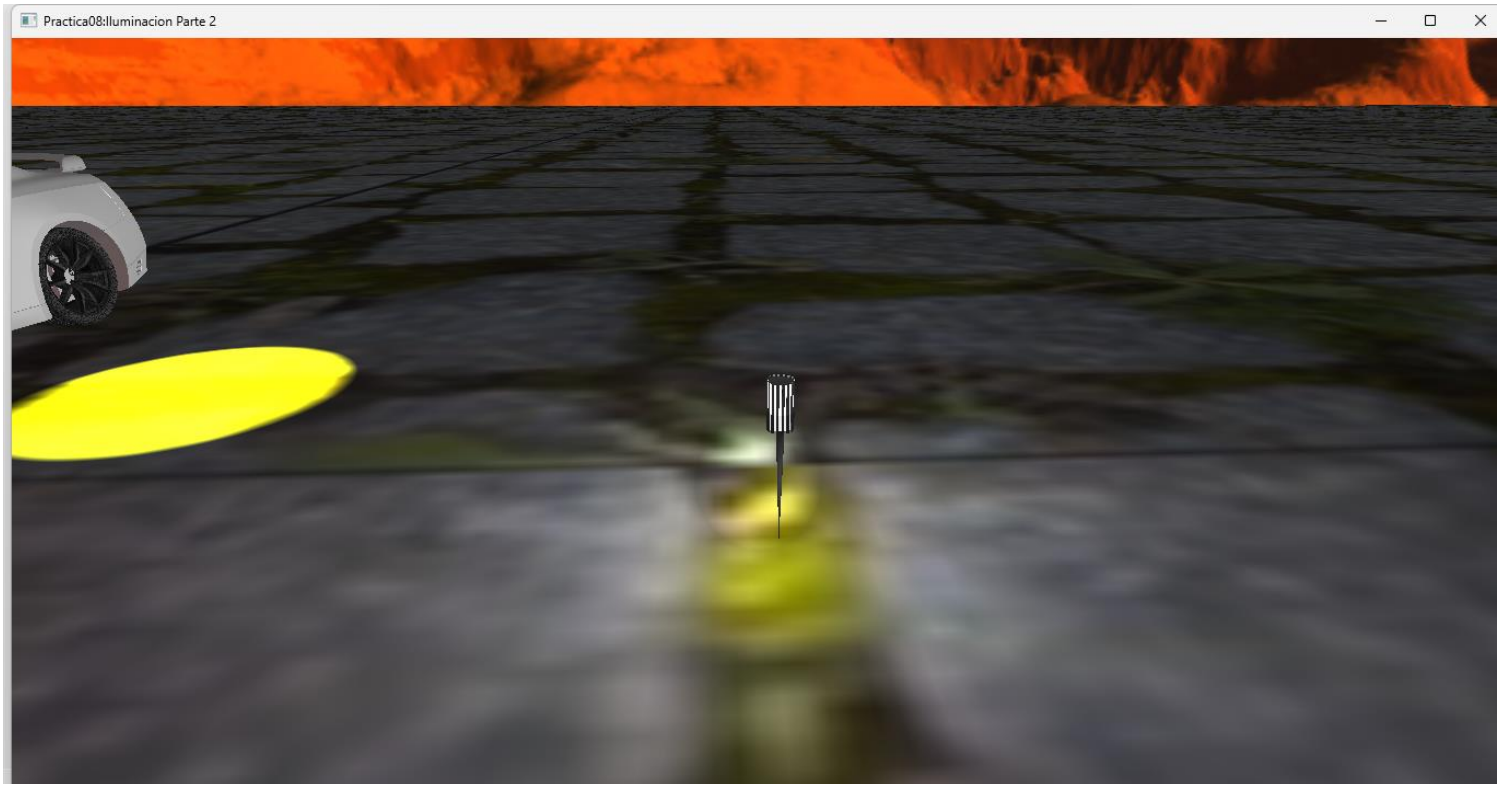
Después de texturizar nuestra lámpara vamos a ejecutar nuestro código.



Apretando la tecla Z



Apretando la tecla X



De igual forma, se adjuntará un vídeo donde se muestre que en tiempo de ejecución se puede prender y apagar la luz de la lámpara por medio del teclado.

Lista de problemas presentados

- Para esta práctica no hubo ningún problema.

Conclusión

Al principio del ejercicio sentí que iba a estar demasiado complicado el ejercicio que consistía en apagar la luz de nuestra lámpara, debido a que al principio si tenía planteado hacer una función de tipo booleana más que nada por simplicidad, pero no sabía como plantear dicha función y me salían errores, pero se solucionaron cuando declaré la variable “apagar” en private en el archivo Window.h. Algo que también es importante mencionar es que al principio tenía planeado hacer que la luz tomará el color negro, pero como vimos en el laboratorio eso no cuenta como apagar una luz. Al final de cuentas siento que ambos ejercicios están al nivel de la clase porque lo de las normales en los modelos lo vimos con el agave entonces para este ejercicio era hacer algo parecido.