

Informática Gráfica

Práctica 4. Iluminación, materiales y texturas.



**UNIVERSIDAD
DE GRANADA**

Grado en Ingeniería Informática

Jesús Pereira Sánchez

jesuspereira@correo.ugr.es

Índice

1	Introducción	2
2	Creación de la escena e Iluminación	2
2.1	Encendiendo DirectionalLight3D	2
2.2	Encendiendo OmniLight3D	3
2.3	Encendiendo SpotLight3D	4
3	Materiales	5
4	Texturas	6
4.1	Texturas en un objeto y mapa de normales	6
4.2	Texturas en los planos	7
5	Texturas en el objeto de revolución	7

1. Introducción

En esta práctica 4, trabajaremos con distintas fuentes de iluminación. Además, crearemos distintos materiales que aplicaremos a los meshes de nuestra escena, y veremos los distintos efectos de brillo que ocurren en estos mismos debido a las fuentes de luz creadas.

También se trabajará con texturas y su aplicación sobre distintas superficies.

2. Creación de la escena e Iluminación

Una vez creada la escena, y añadidas las tres diferentes luces, nos resulta en lo siguiente:

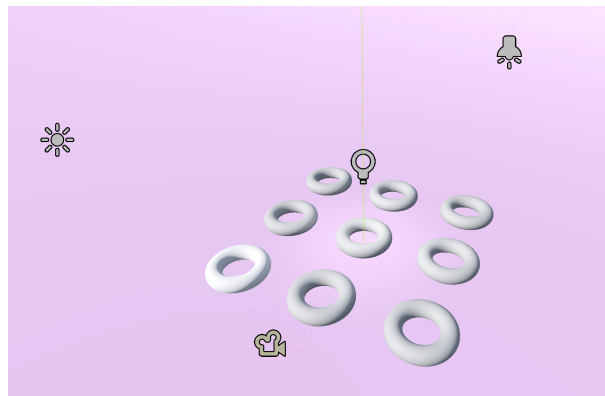


Figura 1: Escena sin luces activadas

Como podemos ver, las luces están añadidas, pero no están activadas puesto que no producen ningún tipo de sombra en los torus que tenemos en la escena. Procedemos a ir encendiendo una en una las luces, a ver que sucede.

2.1. Encendiendo DirectionalLight3D

Al encender la luz direccional, podemos ver como afecta a todos los objetos de la escena, creando la misma sombra para todos, ya que esta luz simula una fuente de iluminación muy grande. Como si fuese el sol.

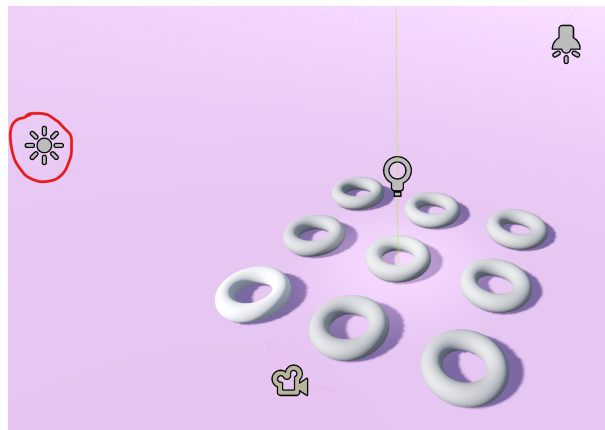


Figura 2: Escena con la luz direccional encendida

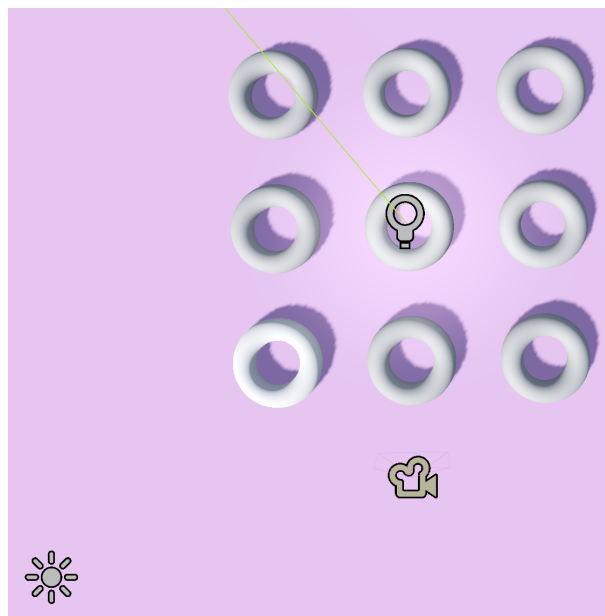


Figura 3: Escena mostrando la misma sombra para todos los objetos

2.2. Encendiendo OmniLight3D

Cuando encendemos la luz puntual, esta estará encima de los objetos, centrada. Esta luz simula una bombilla (iluminación en un punto que alumbra en todas las direcciones), por lo que es de esperar que la luz origine sombras distintas en los distintos objetos que se encuentran a distintas distancias de este punto. Obtenemos lo siguiente:

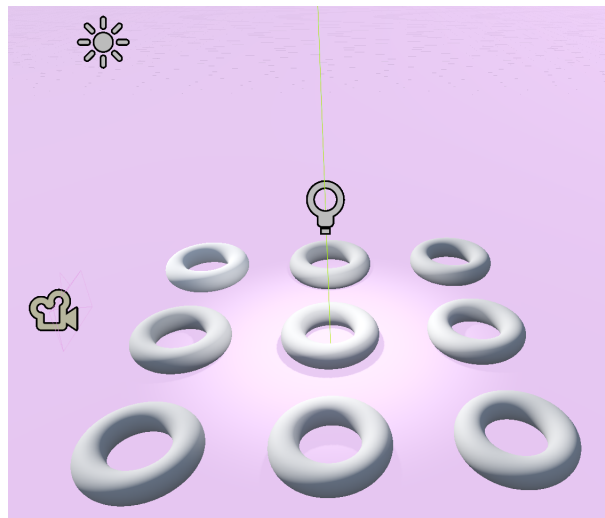


Figura 4: Escena con la luz puntual

Cabe destacar que se ha modificado el valor de *atenuación* de la fuente de luz para poder notar la fuente de luz correctamente.

2.3. Encendiendo SpotLight3D

Ahora encendemos la luz focal, Esta luz se caracteriza por tener un radio y una longitud de acción, puesto que al final simula un foco, un cono que tiene una altura (longitud, rango) y una base (definida por el ángulo de acción). En la imagen podemos ver como se iluminan solo los objetos (y solo parte de ellos) los cuales están dentro del cono de iluminación de este foco.

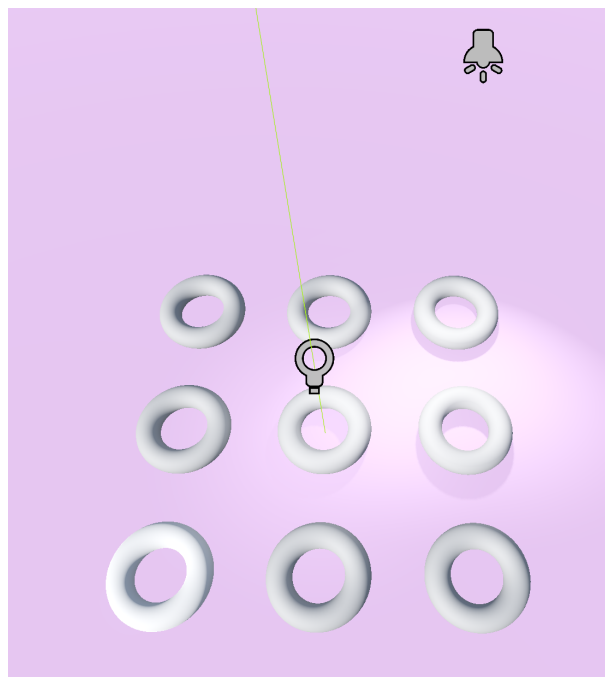


Figura 5: Escena con la luz focal encendida

3. Materiales

Ahora, vamos a aplicar distintos materiales a los objetos que tenemos. Como vemos en la imagen, cada fila tiene un color. Cada columna, tiene una propiedad del material distinta.

1. **Metallic.** Los objetos de esta columna tienen el valor Metallic fijado a 0'75. El Roughness está fijado a 0'15, para simular un metal más realista. Si estuviese fijado a 0 el Roughness, tendríamos metales perfectos sin defectos.
2. **Transparency.** Los objetos de esta columna tienen la Transparencia activada. Para ello, en el color albedo, modificamos el valor alpha, y mostramos la transparencia.
3. **Roughness.** El Metallic está fijado a 0 para poder denotar bien el roughness.

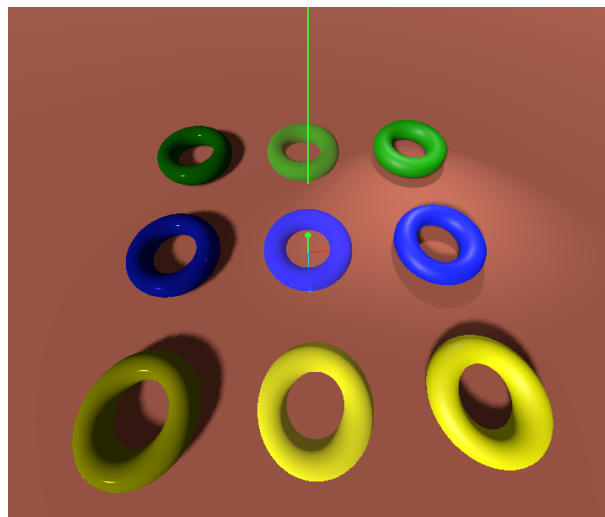


Figura 6: Escena con los materiales aplicados

Las configuraciones del material se pueden guardar en nuestro sistema de archivos, para poder agregarlo a cualquier objeto rápidamente tan solo con arrastrarlo.



Figura 7: Materiales en archivos

4. Texturas

4.1. Texturas en un objeto y mapa de normales

Ahora aplicamos la textura de los ladrillos en el objeto. Se han aplicado también el mapa de normales, poniendo la imagen de mapa de normales en su respectivo canal.

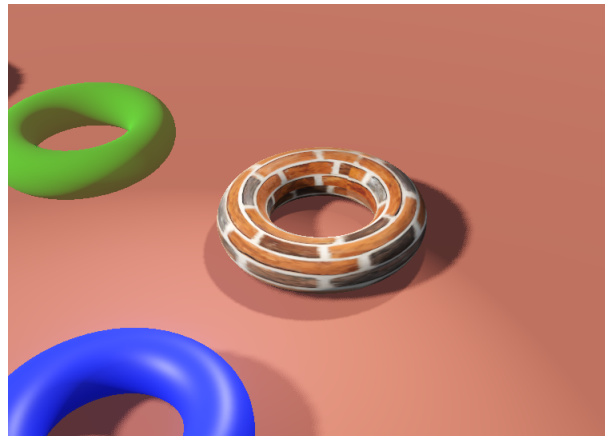


Figura 8: Textura de ladrillos en el objeto

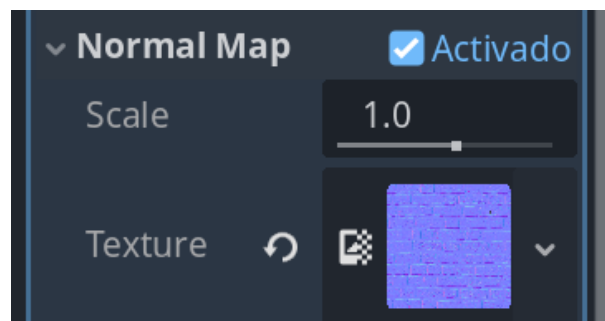


Figura 9: Mapa de normales de los bricks

4.2. Texturas en los planos

Usando el script, creamos los planos, a los cuales le damos la Textura y el Mapa de normales correspondiente.

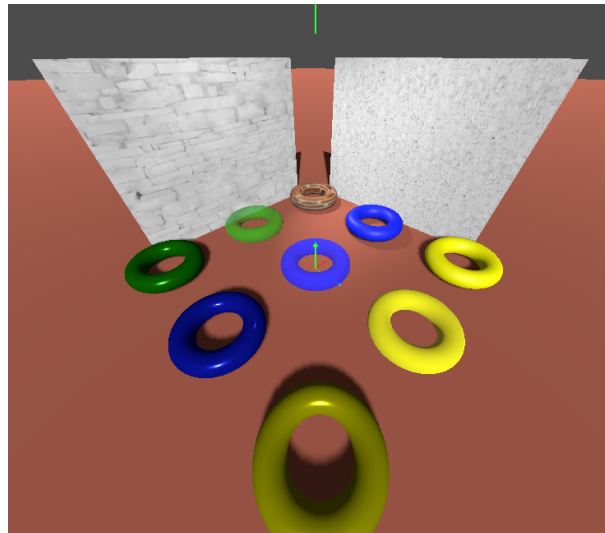


Figura 10: Texturas en los planos

5. Texturas en el objeto de revolución

Para poder aplicar una textura en nuestro objeto de revolución, tenemos que definir las coordenadas de textura. Para ello, se nos da un algoritmo. El algoritmo calcula las coordenadas de textura de nuestro objeto de revolución en función del número de vértices que tenga el objeto. Para cada vértice (x, y, z) del objeto corresponde un vértice (x', y') en la textura.

Las coordenadas del mapa de texturas son de la forma (x', y') , donde x' e y' son valores entre 0 y 1. Para el cálculo de la x , calculamos el desplazamiento con el arcotangente de las coordenadas z x , y y lo transformamos para que se encuentre en el rango que debe. Para la y , usamos una regla de 3 para escalar la y de cada vértice.

El código usado es el siguiente:

```
1 static func calcUV(vertices: PackedVector3Array) -> PackedVector2Array:
2     var uvs := PackedVector2Array()
3     var max_u = 1.0
4     var max_v = 1.0
5     var min_y = vertices[0].y # Bottom point
6     var max_y = vertices[1].y # Top point
7     var height = max_y - min_y
8
9     for vert in vertices:
10        # 1. Calcular el valor del parametro u
11        var phi = atan2(vert.z, vert.x)
12        var u = max_u*(phi / (2.0 * PI) + 0.5) # This step is to make range from 0 to 1
13
14        # 2. Calcular el valor del parametro v
15        var v = (vert.y - min_y) / height
16
17        var uv_coords = Vector2(u, v)
18        uvs.append(uv_coords)
19    return uvs
```

Listing 1: calcUV.gd

Ahora, seleccionamos la textura, y la aplicamos a nuestro objeto de revolución:

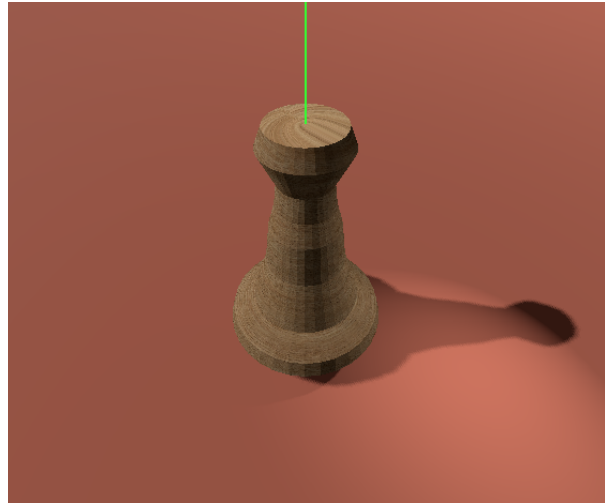


Figura 11: Peón con textura