Tarea 4

Shaders: La Tarea

Profesora: Nancy Hitschfeld e Iván Sipirán
Auxiliares: Julieta Coloma y Vicente González

1. Introducción

Esta tarea se centrará en la implementación de shaders y luces. La escena contará de 5 planetas dibujados con distintos shaders y un sol.

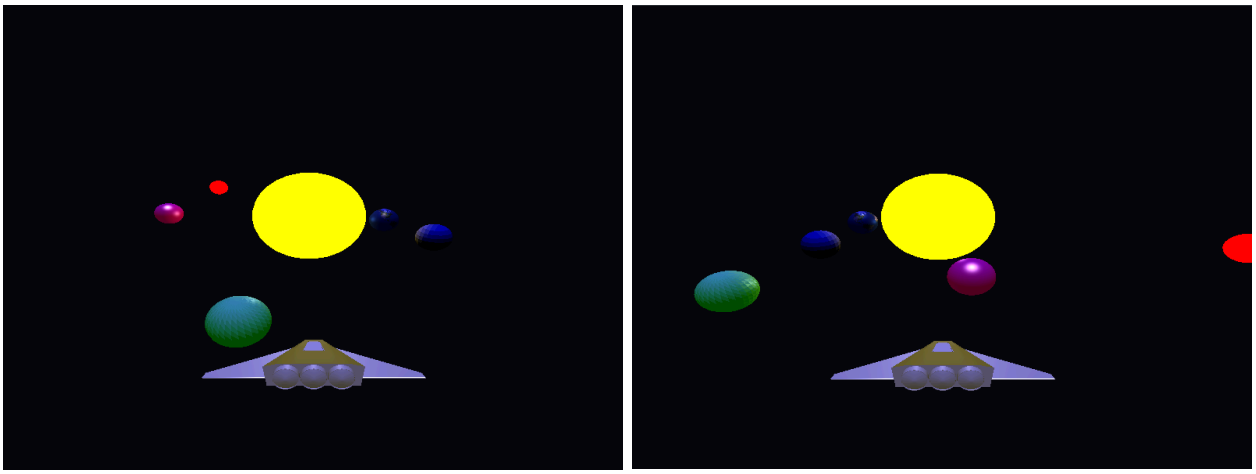

Figura 1: Resultado final.

2. Requisitos

2.1. Shaders (3.5 pts)

Para la tarea tendrá que hacer 5 shaders:

a) Color Shader
b) Flat Shader
c) Phong Shader
d) Toon Shader
e) Textured Shader

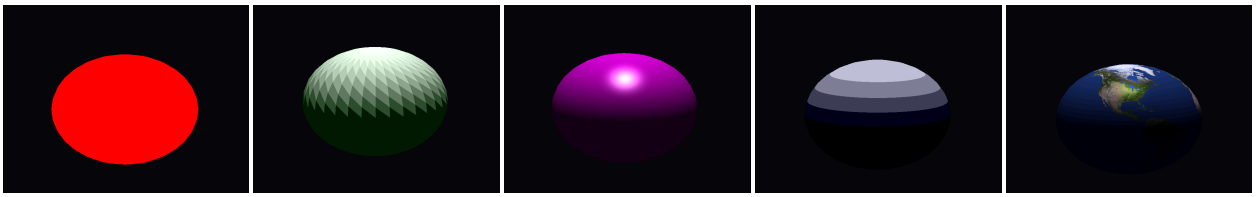

Figura 2: Shaders en el orden mencionado.

2.1.1. Indicaciones varias

- El Color shader no toma en consideración las luces de la escena, solo entrega un color.
- El Phong shader fue estudiado en clases y pueden obtenerlo de los auxiliares.

- El Textured shader también fue estudiado en clases y en auxiliares. Mientras su shader pueda procesar una textura será considerado correctamente como un *textured shader*.

2.1.2. Flat Shader

La principal característica de este shader es que no realiza interpolación para las normales, como resultado cada cara tiene una misma normal y por lo tanto, la misma intensidad de luz.

Para este shader se tiene que tomar en cuenta el efecto de la luz y calcular los componentes **ambiental** y **difuso**.

Para evitar la interpolación de las normales se utiliza la palabra clave «`flat`» en el *vertex shader* y en el *fragment shader*.

```
#en el vertex shader
flat out vec3 normal;

#en el fragment shader
flat in vec3 normal;
```

Esto hará que esa variable no se interpole.

2.1.3. Toon Shader

A continuación algunas indicaciones para hacer un toon shader. El toon shader es un programa que pretende simular una apariencia 2D o *toon* pero en un objeto 3D.

Para ello se debe **discretizar** los valores que puede tener la iluminación, en vez de presentar una iluminación continua se separan claramente las zonas iluminadas de las zonas menos iluminadas.

En el Phong shader teníamos que calculamos el componente difuso de la siguiente manera:

```
float diff = max(dot(normal, light.direction), 0.0f);
vec3 diffuse = light.diffuse * (diff * u_material.diffuse);
```

- Usamos el producto punto para obtener el porcentaje de luz que hay que retornar en base a que tan cerca esta la normal de la dirección de la luz.
- Usamos `max` para no tomar en cuenta los valores negativos.
- Finalmente obtenemos el vector multiplicando los colores originales del material con la intensidad de la luz que acabamos de calcular.

En base a este modelo conocido, usted debe modificar el cálculo de la **intensidad de la luz** tal que se discretice el intervalo de 0 a 1 en N posibles niveles. Por ejemplo, si $N = 4$ entonces los valores posibles para la intensidad de la luz serían 0, 0.25, 0.50 y 0.75.

Además, si estudia el Phong Shader se dará cuenta que se procesan 3 tipos de luces (Directional, Pointlight y Spotlight). En esta tarea **no trabajaremos con spotlights**, por lo que no es necesario incluir ese cálculo en sus shaders.

2.2. Escena (2.5 pts)

2.2.1. Objetos

a) Deben haber 5 planetas y un sol. Pueden elegir cualquier modelo que quieran para representar estos objetos. Con la tarea viene un modelo esférico que pueden usar.

b) Los planetas deben orbitar el sol.
c) Cada planeta debe estar renderizado con un **shader distinto**. Un planeta para cada shader de los mencionados en la sección anterior. *Hint: Deberá crear varios pipelines.*
d) Puede usar cualquier shader para el sol, mientras este se note (*no lo pinte de negro que no se va a ver*).

2.2.2. Luces

a) Su escena debe tener **una luz direccional** con la orientación que usted desee.
b) Su escena debe tener **2 pointlights**:
   - Uno en el Sol
   - Uno en la nave (debe seguir la posición de la nave)
c) **Ambas luces deben afectar a todos los planetas**. *Hint: En el auxiliar 7 se juntaron varios pipelines en uno, ¿qué efecto provoca eso?*

2.3. Nave y Cámara

Esta sección no lleva puntos. Se le entregará un template con una cámara y una nave. No es obligación utilizar el template.

3. Entregable

En esta tarea se adoptará una restricción nueva para la entrega (para ayudar a la correción).

Junto al enunciado viene un zip con todos lo archivos para realizar su tarea. Descomprima el zip y trabaje en esa carpeta.

Cuando quiera entregar comprima la carpeta y suba el zip. Esto es para que los ayudantes no tengan que luchar con correr su tarea. Se ratifica que si su tarea no puede ser ejecutada no se evaluará.

4. Consideraciones

Debe tener en cuenta lo siguiente:

- El plazo de entrega es inamovible.
- El trabajo es individual.
- **No** está permitido el plagio del trabajo de sus compañerxs.
- **Si** está permitido reutilizar cualquier código visto en auxiliares u en otras tareas.

5. Bono

Se recibirá un bono de hasta 0.1 puntos en el promedio final de tareas a quien haga alguna de las siguientes opciones:

- Implementar Rim Lighthing
- Crear su propio shader distinto al resto (queda a criterio del equipo docente)
- Añadir Spotlights en la escena e incluirlos en el Toon Shader.

Es necesario documentar el trabajo realizado para acceder a este bono y debe estar claramente señalado en los comentarios dónde se realizan los cambios que son parte del mismo. **Recuerden que el criterio del bono queda a juicio del cuerpo docente.**