

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

11-11-2015

Práctica Obligatoria 2

Informática Gráfica

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and curve upwards and to the right.

Sebastian Guisado Barreras y Susana Pineda De
Luelmo

1. Ilumina el objeto con al menos dos fuentes de luz:

Para iluminar el objeto con otra fuente de luz lo primero que tenemos que hacer es declarar este nuevo foco al igual que teníamos el anterior. Vamos a iluminar el objeto mediante una luz difusa, este componente de la luz proviene de una dirección y se refleja en todas las direcciones y es independiente de la posición del observador, en nuestro caso la cámara.

Lo primero que tenemos que hacer es declarar los valores de esta nueva fuente de luz, es decir la intensidad y la posición que va a tener nuestra luz. Esto lo haremos mediante las siguientes líneas de código:

```
vec3 I12 = vec3 (0.0, 0.0, 1.0);  
vec3 P12 = vec3 (5.0, 0.0, 0.0);
```

El valor que indiquemos en I1 (intensidad de la luz) será el color de nuestra fuente de luz representado mediante un vector de 3 coordenadas (RGB) en nuestro caso, para diferenciarlo del foco antes creado le hemos dado un valor con el cual nos mostrara un foco de luz de color azul.

Una vez que tenemos los valores de la luz difusa definidos pasamos a añadir al conjunto de luces que tenemos en nuestra escena la segunda luz difusa. Para calcular la iluminación difusa necesitamos la intensidad de la fuente (declarada anteriormente) el coeficiente de reflexión difusa, (depende del objeto y obtiene el valor del color de la textura), la normal de la superficie en el punto P (depende del objeto) y el vector de incidencia de la luz L.

Lo primero que tenemos que hacer es calcular L, para ello utilizamos el siguiente fragmento de código:

```
vec3 L2 = normalize(vec3(view*vec4(P12, 1.0)-pos));
```

De esta forma pasamos la posición del foco a coordenadas de la cámara y le restamos la posición del objeto. Como la matriz view y la posición del foco son vectores de diferente dimensión tenemos que añadir una cuarta coordenada a la posición del foco, la cual nos indicara si se trata de un punto o de un vector, una vez multiplicadas las dos matrices necesitamos volver a tener un vector de 3 coordenadas para poder restarlo a la posición, una vez restadas las dos posiciones en coordenadas de la cámara normalizamos el vector y obtenemos L.

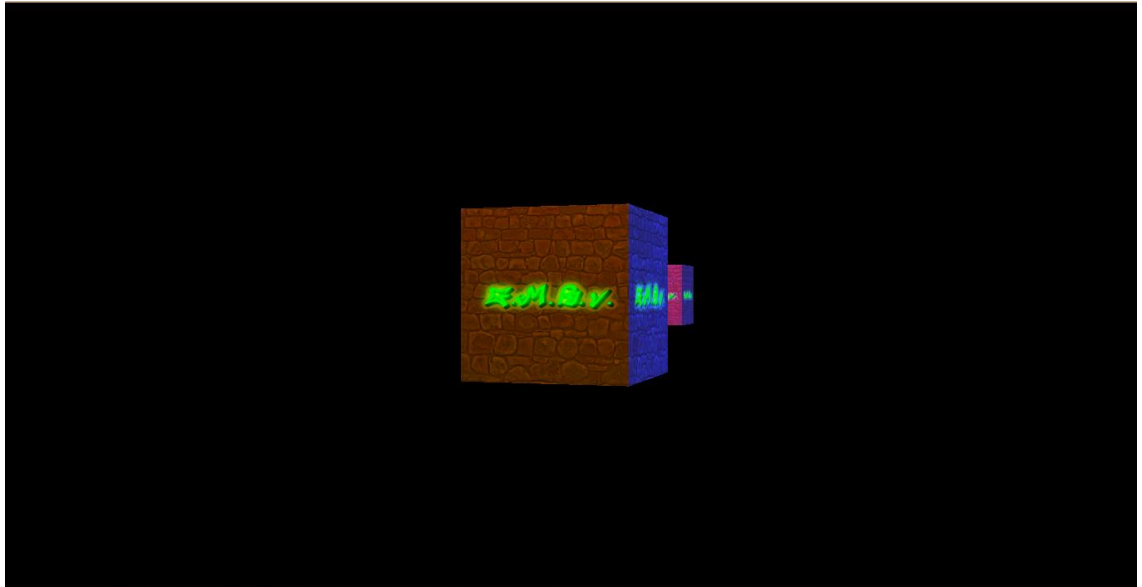
Una vez que contamos con todos los datos necesarios solo tenemos que calcular la iluminación difusa de nuestro nuevo foco y añadirlo al resto de luces de la escena. Para ello utilizamos la siguiente sentencia:

```
color += I12*Kd*dot(N,L2);
```

Por ultimo tenemos que añadir la luz especular que se generara con el nuevo foco de luz. Para ello vamos a pasar a definir la nueva luz especular:

```
vec3 V2 = normalize(-pos);  
vec3 R2 = normalize(reflect(-L2,N));  
float factor2 = clamp(dot(V2,R2),0.0001,1.0);  
factor2 = pow(factor2,n);  
color += I12 * Ks * factor;  
color += Ke;
```

De esta manera ya tenemos nuestro objeto iluminado con dos fuentes de luz. A continuación podemos observar el objeto iluminado con dos fuentes de luz difusas, una de color rojo y una de color azul. En los puntos en los que estas dos luces coinciden se forman los colores correspondientes a estas mezclas en RGB.



NOTA: las capturas de pantalla no coinciden con los objetos realmente implementados en las prácticas por problemas al renderizar de uno de los ordenadores empleados en las prácticas.

2. Añade una nueva funcionalidad que atenúe la intensidad lumínica en función de la distancia del objeto a la fuente lumínica.

Para hacer la atenuación de la luz en función de la distancia del foco de luz al objeto contamos con la siguiente formula:

$$fatt = \min\left(\frac{1}{c_1 + c_2d + c_3d^2}, 1\right)$$

El factor de atenuación *fatt* será incluido en el cálculo de la intensidad de la luz que queramos atenuar multiplicándolo por dicha intensidad.

Ci serán constantes definidas por el usuario para cada fuente.

Vamos a calcular el factor de atenuación para las dos luces creadas anteriormente. Para ello lo primero que tenemos que hacer es definir las variables que nos faltan (*ci*). Crearemos dos vectores *C* y *C2* que nos indicaran los valores de *C1*, *C2*, *C3* de nuestra fórmula para cada uno de los dos focos.

```
vec3 C = vec3 (1.0, 0.1, 0.0); //Luz 1
vec3 C2 = vec3 (1.0, 0.1, 0.0);
```

Para comprobar los resultados pondremos los dos factores de atenuación iguales y las posiciones de los focos diferentes para poder observar la diferencia entre una luz cercana y una lejana.

Una vez tenemos los coeficientes necesitamos la distancia de la luz al objeto, la cual la obtenemos mediante el siguiente fragmento de código:

```
float d = length (P1-pos); //Distancia de la primera luz al
objeto
float d2 = length (P2-pos); //Distancia de la segunda luz al
objeto
```

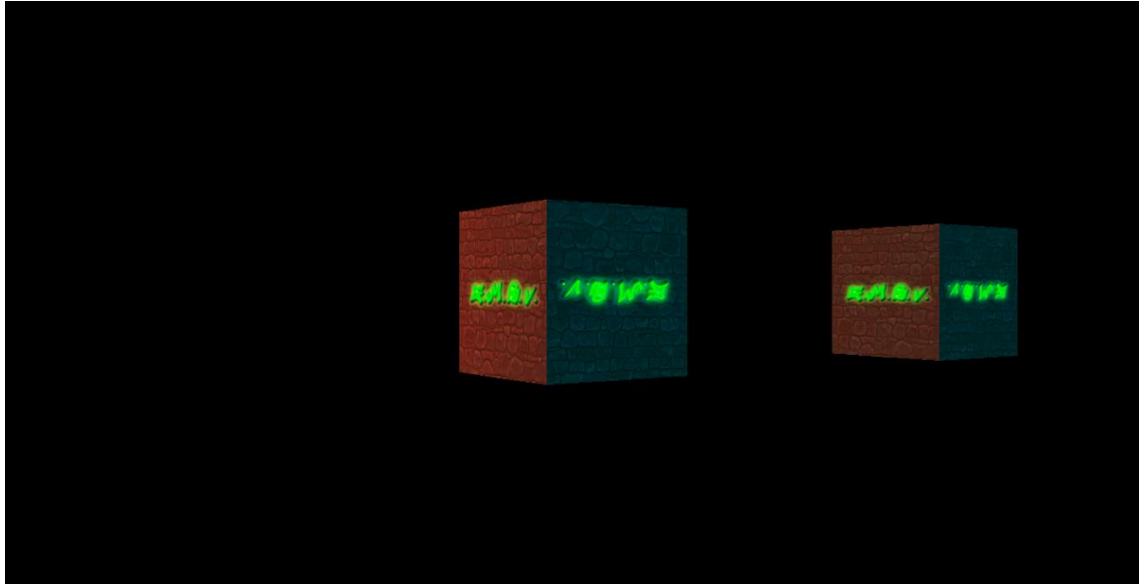
Una vez que tenemos todos los datos solo tenemos que implementar la fórmula de la atenuación:

```
float fatt = 1/ (C.x + C.y*d + C.z*pow(d, 2)); //fatt luz1
float fatt2 = 1/ (C2.x + C2.y*d2 + C2.z*pow(d2, 2)); //fatt luz2
```

Ahora que ya tenemos calculados los factores de atenuación lo único que tenemos que hacer es aplicarlos a la intensidad de cada una de las fuentes de luz.

```
color += fatt*I1*Kd*dot(N,L); //Intensidad luz1
color += fatt2 * I2*Kd*dot(N,L2); //Intensidad luz2
```

A continuación podemos ver como la luz proveniente de la fuente de luz roja es mucho más marcada que la proveniente de la fuente de luz azul. De esta forma podemos comprobar que dependiendo de la distancia de la fuente de luz recibimos más o menos intensidad de esa fuente.



NOTA: las capturas de pantalla no coinciden con los objetos realmente implementados en las prácticas por problemas al renderizar de uno de los ordenadores empleados en las prácticas.

3. Implementa los siguientes tipos de luz: luz direccional, luz focal.

Vamos a comenzar implementando la Luz direccional. Las luces direccionales son aquellas que están situadas en el infinito por lo que los haces de luz son paralelos. Para ello lo primero que vamos a hacer es declarar los valores de la intensidad y la posición de la nueva luz. Para ello contamos con el siguiente fragmento de código:

```
vec3 I13 = vec3(0.0, 1.0, 0.0);  
vec3 P13 = vec3(2.0, 0.0, 0.0);
```

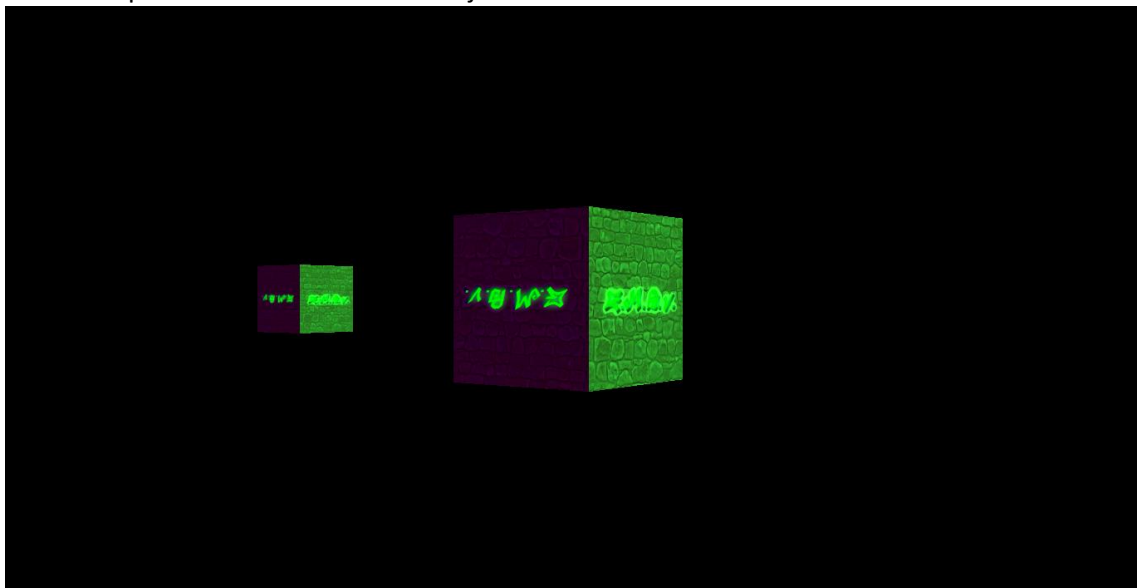
A continuación lo único que tenemos que hacer es calcular la L de la posición tomando la posición del foco (P13) como un vector, cuarta coordenada del P13 a 0.

```
vec3 L3 = normalize(vec3(view*vec4(P13, 0.0)-pos));  
color += I13*Kd*dot(N,L3);
```

Una vez que tenemos esta información solo nos queda calcular la luz especular de esta fuente de luz de la siguiente forma:

```
vec3 V3 = normalize(-pos);  
vec3 R3 = normalize(reflect(-L3,N));  
float factor3 = clamp(dot(V3,R3),0.0001,1.0);  
factor3 = pow(factor3,n);  
color += I13 * Ks * factor;  
color += Ke;
```

Una vez que tenemos calculada L3 y la luz especular asociada a esta fuente de luz solo tenemos que añadir la nueva luz al conjunto de luces de la escena.



NOTA: las capturas de pantalla no coinciden con los objetos realmente implementados en las prácticas por problemas al renderizar de uno de los ordenadores empleados en las prácticas.

A continuación pasamos a implementar la luz focal, para ello, como anteriormente declaramos la intensidad del foco y su posición mediante el siguiente fragmento de código:

```
vec3 I14 = vec3(1.0, 1.0, 1.0);  
vec3 P14 = vec3(0.0, 0.0, 8.0);
```

Las luces focales se implementan como luces puntuales con un ángulo de apertura. Para ello necesitamos tres variables adicionales: D1 que nos indica la dirección del foco, Aper que es el ángulo de corte tras el cual la intensidad será 0, e variable que controlará la atenuación de la luz según nos acerquemos al ángulo de corte.

```
vec3 D1 = normalize(vec3(vec4(-P14, 1.0)));  
float Aper = 3.14159/50;  
float e = 0.5;
```

Lo primero que haremos es calcular la atenuación que sufrirá el foco de luz según nos acerquemos al ángulo de corte. Para ello utilizaremos la siguiente fórmula:

$$I_p = \left(\frac{D1 * (-L) - \cos(Aper)}{1 - \cos(Aper)} \right)^n * Idiff$$

Aplicaremos esta fórmula si $D1 * (-L) > \cos(Aper)$, en caso contrario la intensidad de la luz focal será 0.

Lo primero que haremos será calcular L para este nuevo foco de luz mediante:

```
vec3 L4 = normalize(vec3(vec4(P14, 1.0)-pos));
```

A continuación pasaremos a implementar la condición a partir de la cual iluminaremos nuestro cubo:

Si $D1 * (-L) > \cos(Aper)$ tenemos que calcular el coeficiente de atenuación (Coe) y después aplicarlo a la intensidad del foco. Para ello utilizamos el siguiente fragmento de código:

```
float Coe = pow((dot(vec3(D1), vec3(-L4)) - cos(Aper)) / (1 - cos(Aper)), e);
```

Vamos a calcular también la atenuación según la distancia al foco, para ello haremos lo mismo que en las otras fuentes de luz:

```
vec3 C3 = vec3(1.0, 0.01, 0.0);  
float d3 = length(P14-pos);  
float fatt3 = 1 / (C3.x + C3.y*d3 + C3.z*pow(d3, 2));
```

Una vez calculados los coeficientes de atenuación los aplicamos a la intensidad de la luz del foco y lo añadimos al conjunto de luces de la escena:

```
color += fatt3*Coe*I14*Kd*dot(N,L4);
```

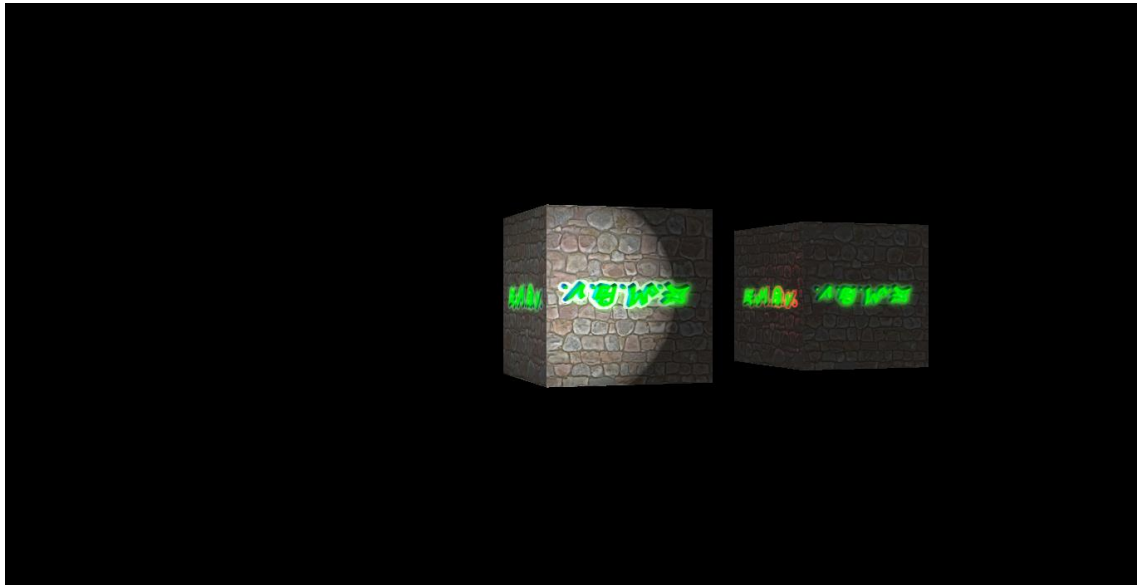
Finalmente pasamos a implementar el caso en el cual $D1*(-L) < \cos(A_{per})$, en el cual lo único que tenemos que indicar es que la intensidad de esta fuente de luz será 0.

```
else {  
color += vec3 (0.0);  
}
```

Por ultimo pasamos a calcular la luz especular relacionada con la luz focal:

```
vec3 V4 = normalize(-pos);  
vec3 R4 = normalize(reflect(-L4,N));  
float factor4 = clamp(dot(V4,R4),0.0001,1.0);  
factor4 = pow(factor4,n);  
color += I14 * Ks * factor;  
color += Ke;
```

De esta forma quedaría implementada la luz focal.



NOTA: las capturas de pantalla no coinciden con los objetos realmente implementados en las prácticas por problemas al renderizar de uno de los ordenadores empleados en las prácticas.