

6. Lighting 5 (lighting5.*) (1^{er} control lab, curs 2011-12, Q2)

Aquest problema és similar a Lighting (4), però es demana que els càlculs d'il·luminació es facin en eye space o world space dependent d'una variable uniform.

Escriu un **vertex shader** i un **fragment shader** que calculi la il·luminació per fragment fent servir el model de Phong. El color C del fragment es calcula en funció dels vectors N, V i L,

$$C(N, V, L) = K_a I_a + K_d I_d (N \cdot L) + K_s I_s (R \cdot V)^s$$

on

N = vector normal unitari en el punt

V = vector unitari del punt cap a la càmera

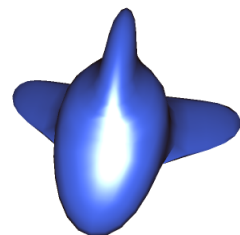
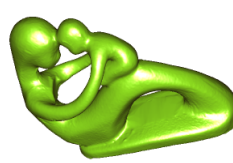
L = vector unitari del punt cap a la font de llum


Per tal de simplificar el problema, us proporcionem aquesta implementació de C(N, V, L), que podeu copiar i pegar al vostre shader i podeu cridar sense cap modificació:

```
vec4 light(vec3 N, vec3 V, vec3 L)
{
    N=normalize(N); V=normalize(V); L=normalize(L);
    vec3 R = normalize( 2.0*dot(N,L)*N-L );
    float NdotL = max( 0.0, dot( N,L ) );
    float RdotV = max( 0.0, dot( R,V ) );
    float Idiff = NdotL;
    float Ispec = 0;
    if (NdotL>0) Ispec=pow( RdotV, matShininess );
    return
        matAmbient * lightAmbient +
        matDiffuse * lightDiffuse * Idiff+
        matSpecular * lightSpecular * Ispec;
}
```

Els shaders rebran una variable **uniform bool world** indicant si els càlculs d'il·luminació s'han de fer en world space o eye space. Si world és fals, el vostre fragment shader haurà de cridar a light(N,V,L) amb N,V i L en eye space. Si és cert, s'haurà de cridar amb N,V i L en world space. **Important:** heu d'assumir que en aquest exercici la transformació de modelat és la identitat; per tant en aquest exercici object space i world space coincideixen.

La imatge resultant en tots dos casos haurà de ser la mateixa.



$(x, x, x) = InvMx(0,0,0)$ Obs
 como el vector V va desde el punto al observado, entonces $V = obs - p$

 P = vertex (punto en el cual quiero calcular la iluminación)