

4. Lighting 2 (lighting2.*)

Escriu un **vertex shader** que calculi la il·luminació per vèrtex amb el **model de Phong**.

La fórmula a aplicar serà la de Phong:

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

color llum ambiental (pointing to $K_a I_a$)
componente difusa (pointing to $K_d I_d (N \cdot L)$)
componente especular (pointing to $K_s I_s (R \cdot V)^s$)

on

K_a, K_d, K_s = reflectivitat ambient, difosa i especular del material

s = shininess del material (*taca*)

I_a, I_d, I_s = propietats ambient, difosa i especular de la llum

N = vector normal unitari

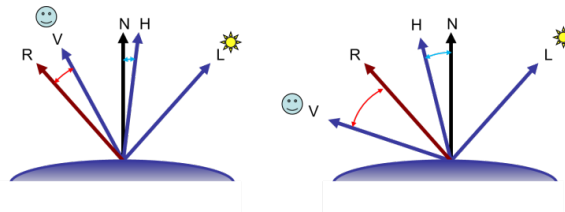
L = vector unitari cap a la font de llum

V = vector unitari del vèrtex cap a la càmera

R = reflexió del vector L respecte N . Es pot calcular com $R = 2(N \cdot L)N - L$

Totes les components haurien d'estar en coordenades de la càmera.

El model de Phong calcula el terme especular a partir del cosinus de l'angle format pels vectors R i V (en vermell a la figura):



Observació: a la fórmula anterior, cal evitar “restar” il·luminació quan els productes escalars $N \cdot L$ o $R \cdot V$ són negatius. Per tant haureu de fer servir $\max(0.0, \text{dot}(N, L))$ i $\max(0.0, \text{dot}(R, V))$. També cal evitar que el terme especular intervingui quan $N \cdot L < 0$, ja que això vol dir que la llum prové del darrera.

Aquí teniu un exemple:

