Lighting Cheatsheet

1 Definitions

incident light : light hitting surface of object
directional light : light with a uniform direction and intensity
point light : light with a location, direction and intensity depend on relative location and distance

2 Light Sources

$\boldsymbol{l_s}$: intensity (RGB) of specular component of light from light source l
$\boldsymbol{l_d}$: intensity (RGB) of diffuse component of light from light source l
$\boldsymbol{l_a}$: intensity (RGB) of ambient component of light from light source l
$\boldsymbol{l_D}$: direction of rays of light, pre-reflection, from light source l(for directional lights)
$\boldsymbol{l_P}$: position of light source l (for point lights)
$\boldsymbol{a}$: intensity (RGB) of ambient light

- uniform for entire scene
- base ambient light for scene plus sum of all l_a from individual light sources

,
$\boldsymbol{r_l}$: "radius" ; distance from light source l to fragment, only applicable to point point lights

3 Materials

$\boldsymbol{m_s}$: specular reflection constant of material m

- ratio of incident light reflected specularly
- may be a single constant, or a $(red, greenblue)$ triple

$\boldsymbol{m_d}$: diffuse reflection constant of material m

- ratio of incident light reflected diffusely
- may be a single constant, or a $(red, greenblue)$ triple

$\boldsymbol{m_a}$: ambient reflection constant of material m

- ratio of ambient light reflected
- may be a single constant, or a $(red, greenblue)$ triple

$\boldsymbol{\alpha}$: shininess constant

- higher for smoother, more mirror-like surfaces
- lower for rougher, duller surfaces
- high α means small, bright specular reflection
- low α means large dim specular reflection

$\boldsymbol{C}$: color (RGB) of surface

- if m_s, m_d and m_a are (RGB) triples, this is redundant and is not used
- for textured models, often m_s, m_d and m_a are single constants, and C is sampled from a texture and used with illumination (defined below) to determine displayed fragment color

4 Spacial Factors

$\boldsymbol{S_l}$: normalized vector pointing toward light source l from surface

- if l is a directional light, S_l is just $-l_D$
- if l is a point light, S_l is the unit vector from the surface toward the light source (i.e. $lightPosition - fragPosition$, normalized)

$\boldsymbol{N}$: surface normal
$\boldsymbol{R_l}$: direction of light from S_l, perfectly reflected off of surface:

$$2(S_l \cdot N)N - S_l$$

$\boldsymbol{V}$: normalized direction from surface to viewer / camera

5 Putting it all together

$\boldsymbol{D}$: set of all directional light sources
$\boldsymbol{P}$: set of all point light sources
$\boldsymbol{I}$: illumination (RGB) on surface:

$$I = m_a a + \sum_{l \in D} \Big(m_d (S_l \cdot N) l_d + m_s (R_l \cdot V)^\alpha l_s \Big) + \sum_{l \in P} \frac{\Big(m_d (S_l \cdot N) l_d + m_s (R_l \cdot V)^\alpha l_s \Big)}{\max(r_l^3, 1)}$$

- if m_s, m_d and m_a are (RGB) vectors, use the Hadamard product to multiply them by other vectors
- otherwise, just use scalar multiplication

$\boldsymbol{FragColor}$: color displayed from illiumination I on surface with color C

- if material components are single constants: $FragColor = (I_r C_r, I_g C_g, I_b C_b)$
- if material components are colors: $FragColor = I$