CSC 240 Computer Graphics Video 14: Lighting and Shading

Nick Howe
Smith College

Some slides & content courtesy Sara Mathieson & Eitan Mendelowitz

*Photo by **Evie Shaffer** from **Pexels***

Lighting & Shading

Lighting: the arrangement of light emitters in a scene

Shading: surface color from light, materials & geometry

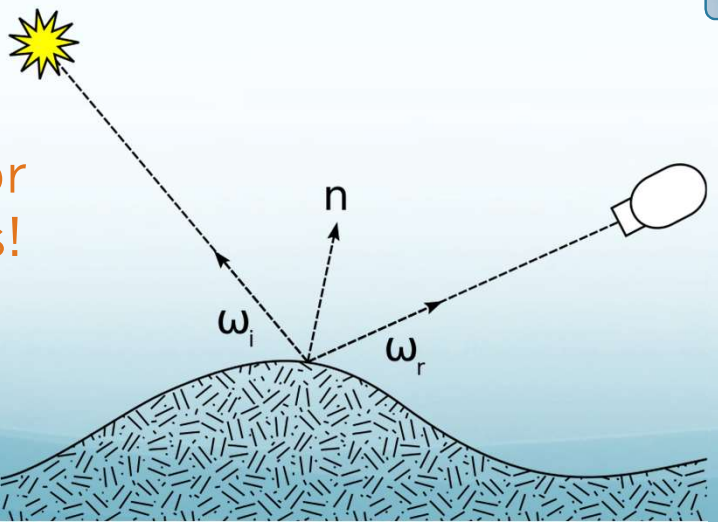
https://www.deviantart.com/sherbertdoesbases/art/Stage-Lights-740575407
https://www.123rf.com/photo_4577004_classic-white-marble-bust-of-athena-isolated-on-black-background.html

Shading

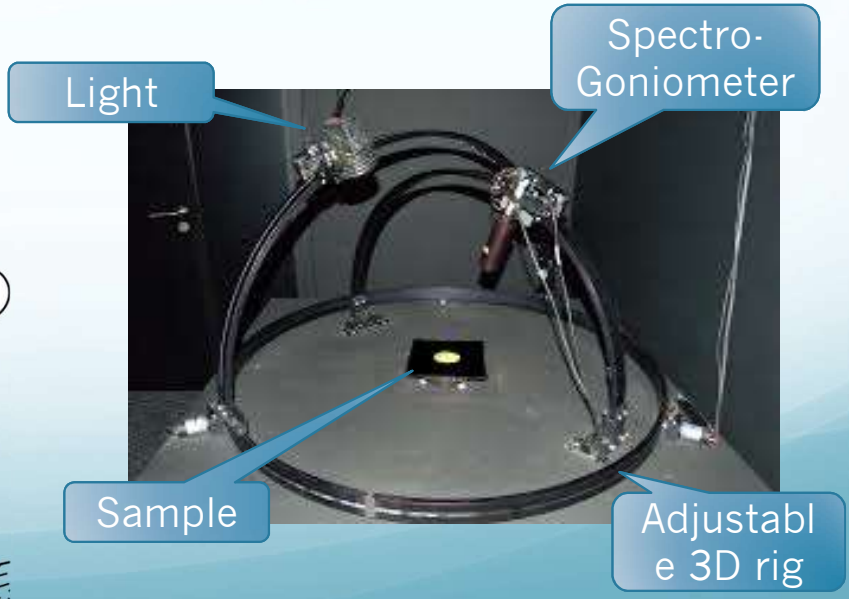
Bidirectional Reflectance Distribution Function (BRDF) specifies how light scatters from a surface material

- Varies with incident angle, outgoing angle, wavelength
- May vary over the surface as well

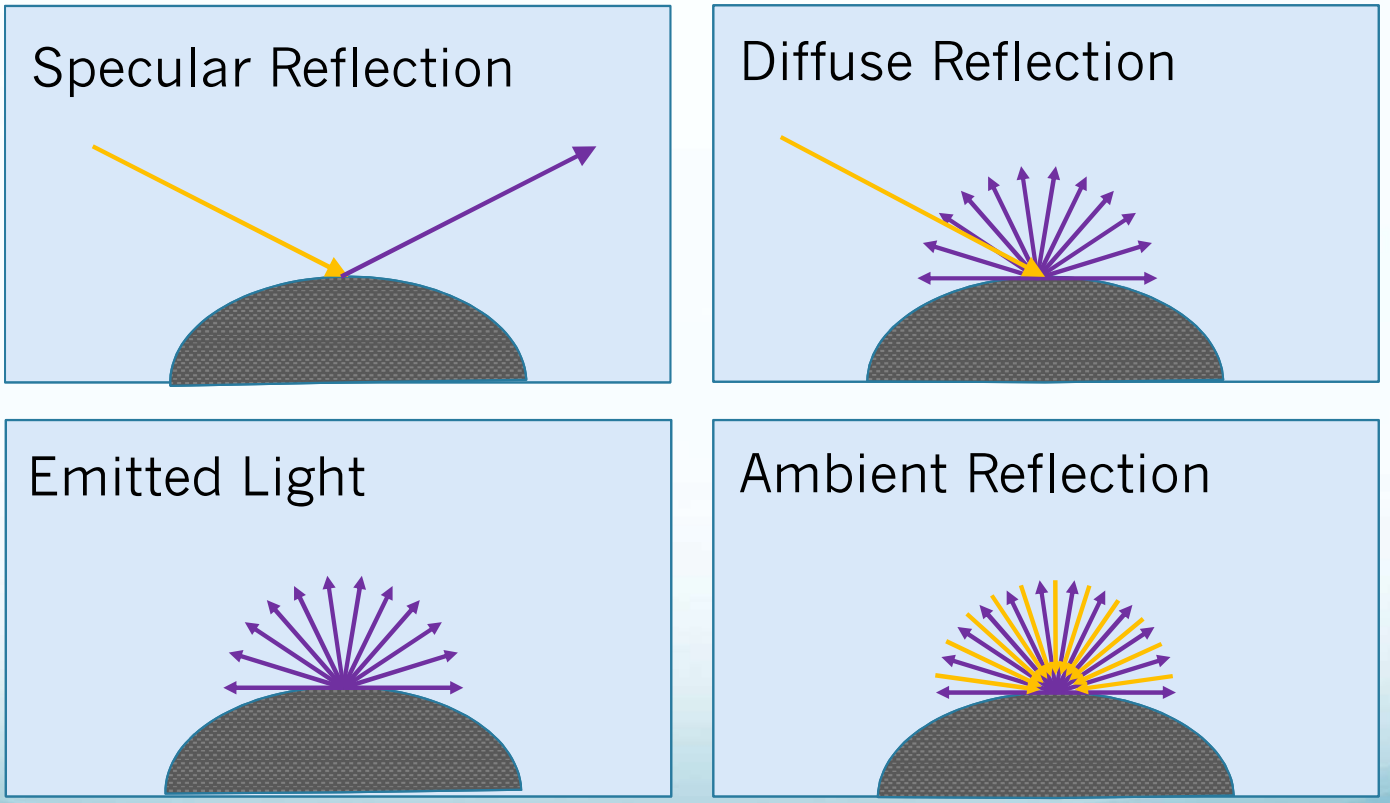
Too complex for most purposes! Must simplify.



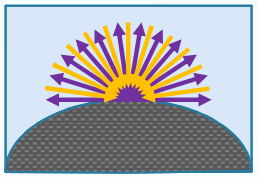
wikipedia



Types of Light Transmission in Three.js



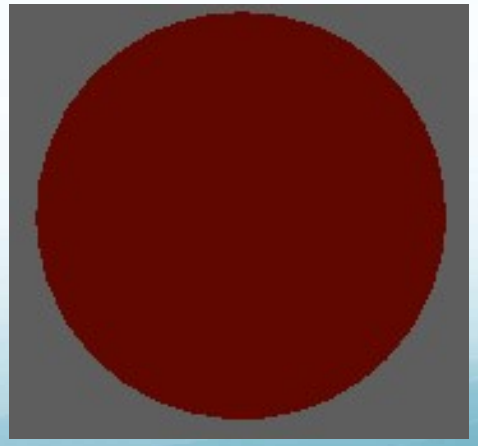
Ambient Reflection



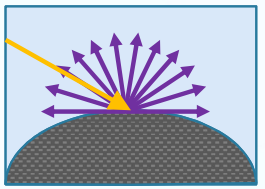
Ambient light affects all surfaces equally and emits equally in every direction

Pixel color = k_a x (material color) x (ambient light color)

- Every point looks alike under ambient reflection
- Shapes don't look 3D!
  - No shadows…
  - No highlights…
  - Surface detail is lost



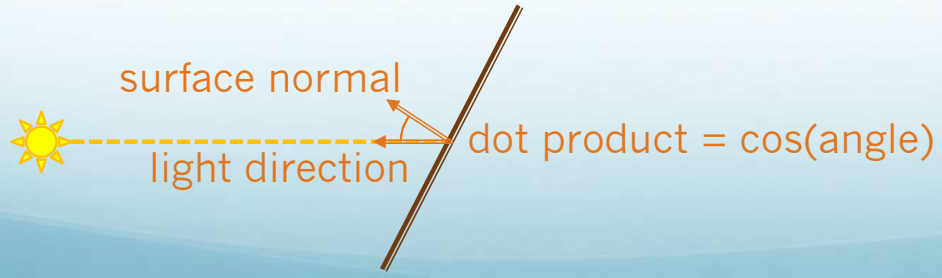
Diffuse Reflection

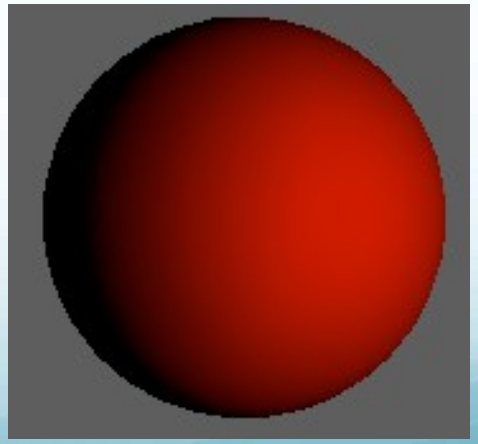


Directional incoming light is scattered in every direction

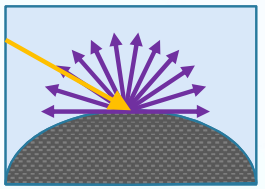
Pixel color = k_d x (material color) x (light color)
x [(light direction)·(surface normal)]

- Points are brighter if they face the light source
- Points are darker if they face away
- Gives 3D effect to rendering





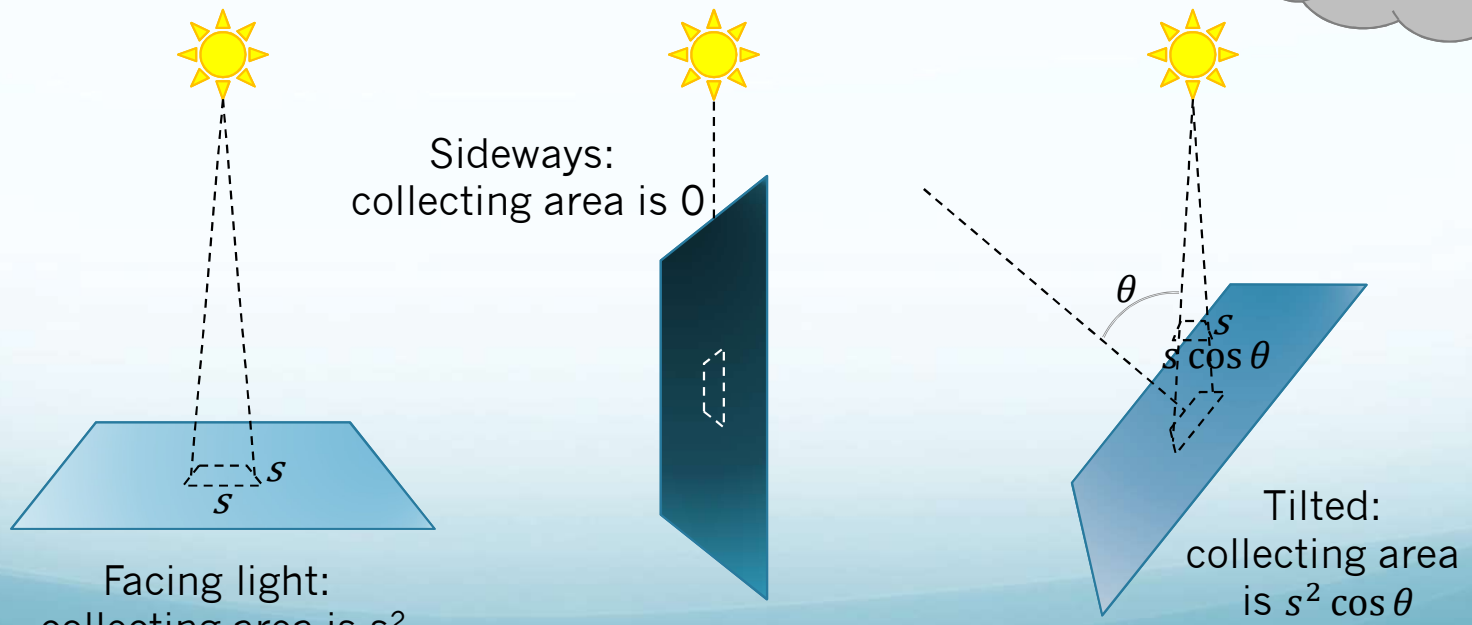
Diffuse Reflection



Why does the light orientation matter for diffuse reflection?

- Simulates the amount of light striking a surface

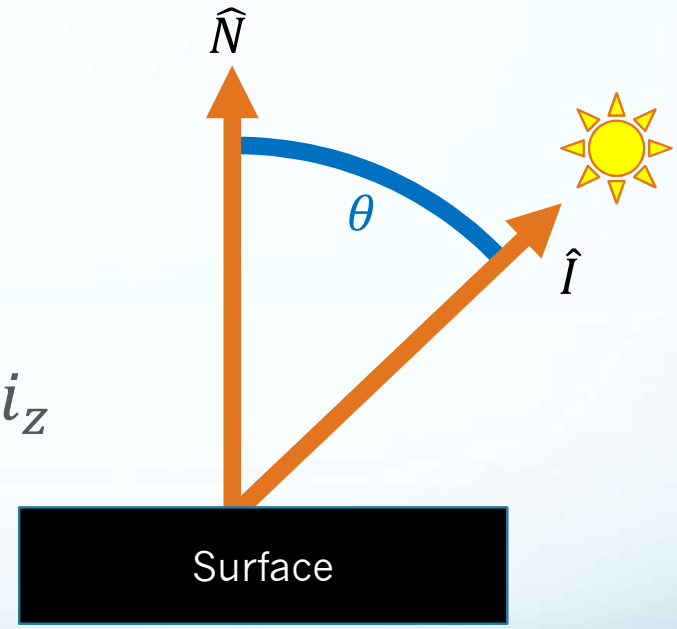
*Also called Lambertian reflectance



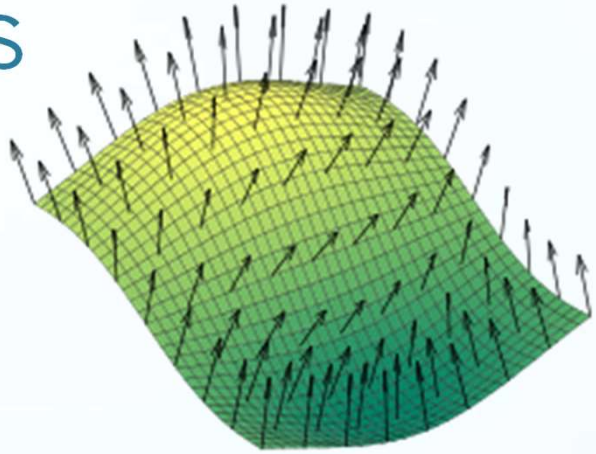
Dot Product

- Normal unit vector $\widehat{N} = (n_x, n_y, n_z)$
- Incidence vector (points to light) $\hat{I} = (i_x, i_y, i_z)$
- Angle between is θ
- Dot product:

$$\widehat{N} \cdot \hat{I} = n_x i_x + n_y i_y + n_z i_z$$
$$= \|\widehat{N}\| \|\hat{I}\| \cos\theta$$
$$= \cos\theta$$



Surface Normals

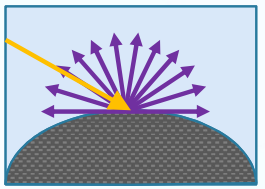

wikipedia

Surface normal vectors represent orientation

- A **normal vector** is scaled to length 1
- Surface normals are **perpendicular** to the surface at each point (right hand rule)
- Curved surfaces may have a different normal at every single point
  - What are the normal vectors of a sphere?
- Graphics system needs to know the surface normal to compute shading
  - Store limited number of normal vectors per polygon face

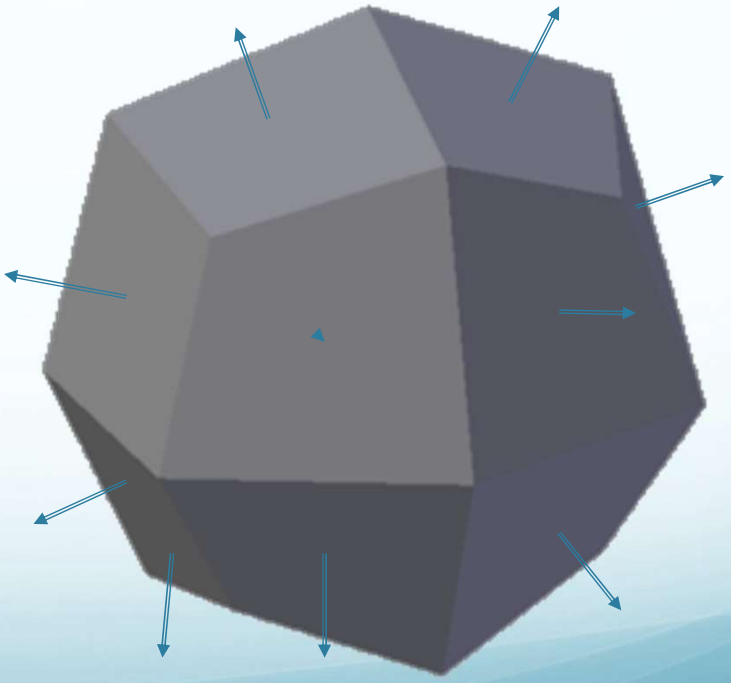
Pixel color = k_d x (material color) x (light color)
x [(light direction)·(surface normal)]

Flat Shading



Simplest shading model is one normal per face

- Low storage requirements
- Entire face gets one coloring, based on normal vector for that face
- Gives faceted look with sharp edges

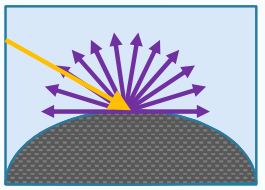


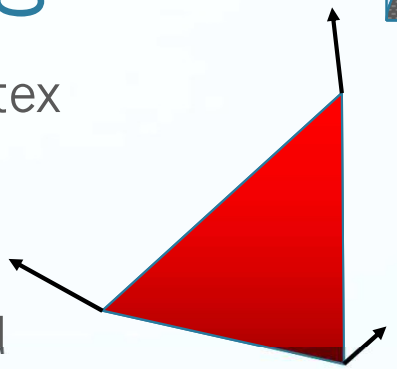
https://computergraphics.stackexchange.com/questions/1888/flat-shading-for-non-planar-polygons

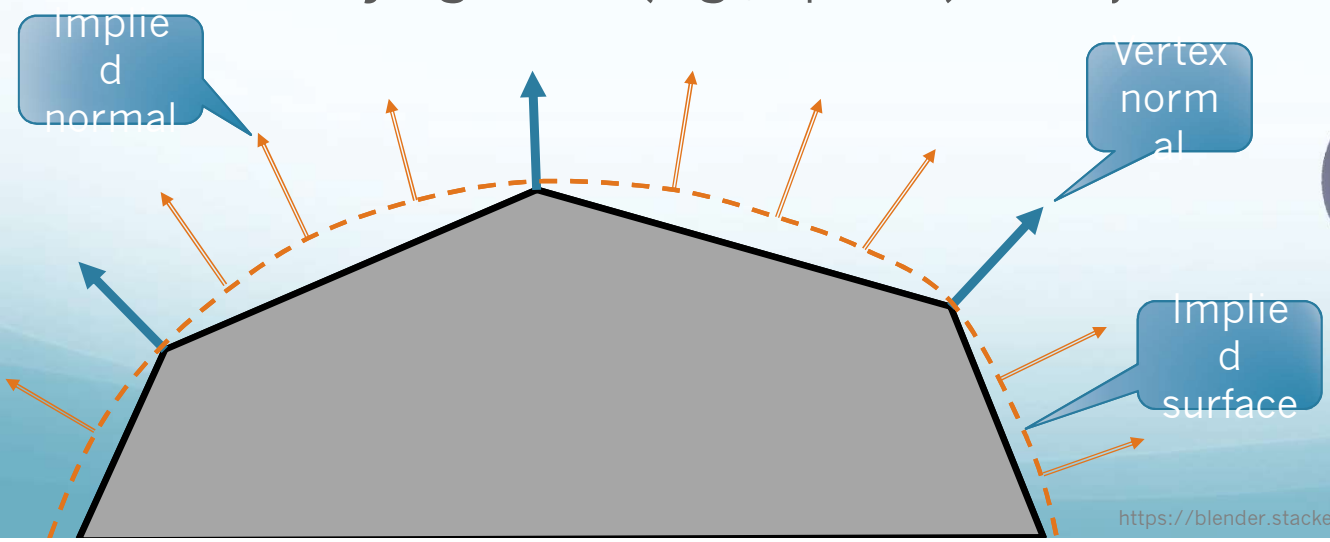
Goraud Shading

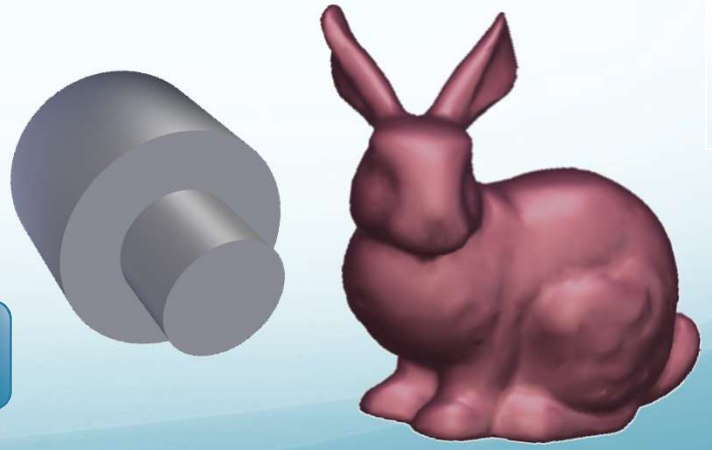
Goraud shading (1971) stores normal at each vertex and interpolates color across the face

- Gives smoother-looking surface
- Vertex normals can be computed from intended underlying curve (e.g., sphere) or adjacent faces



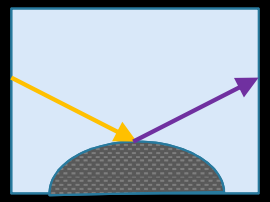




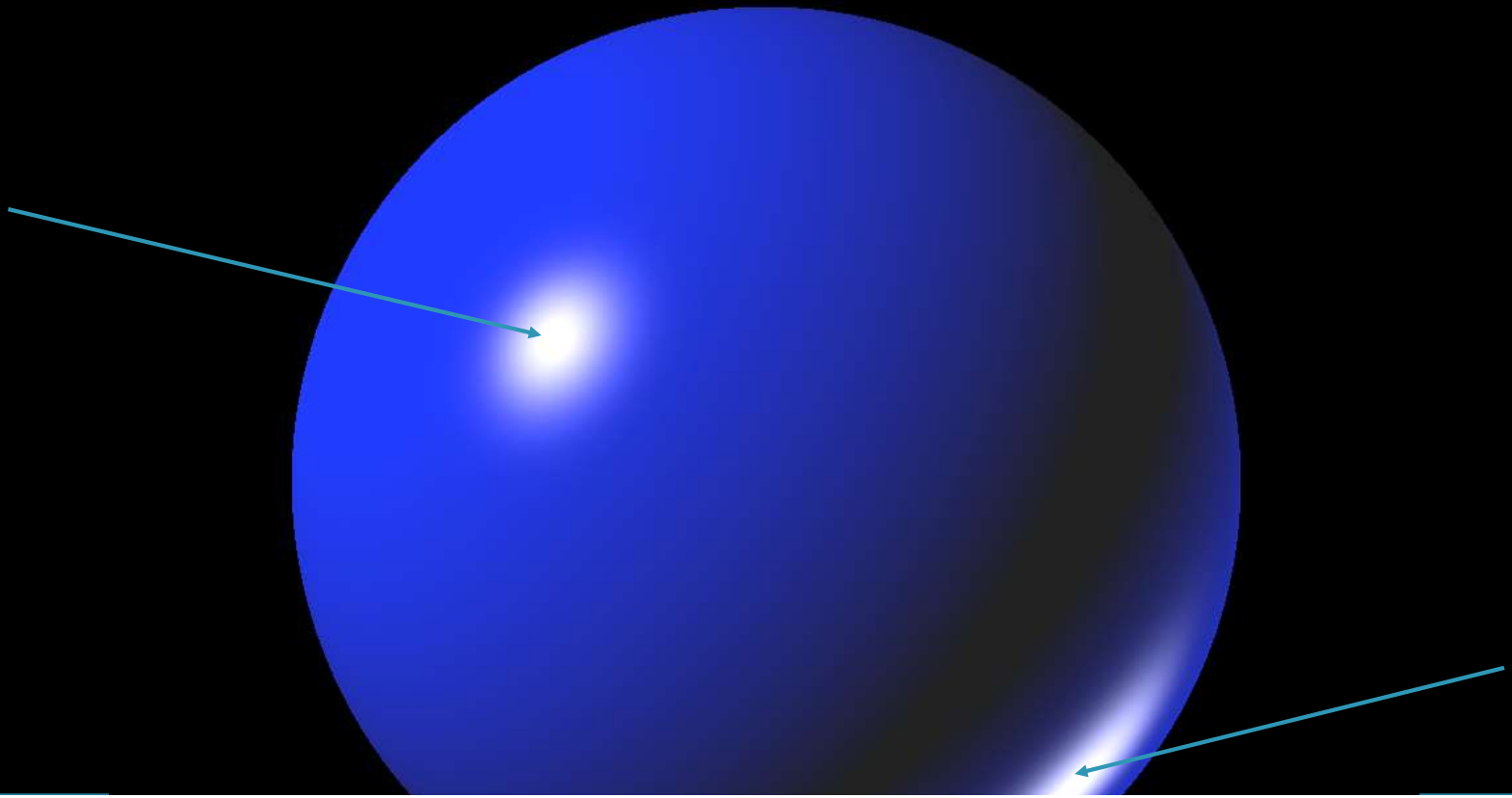


https://blender.stackexchange.com/questions/14627/matcap-smooth-shading-in-blender-2-71-very-bloc

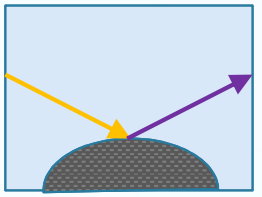
Specular Reflection

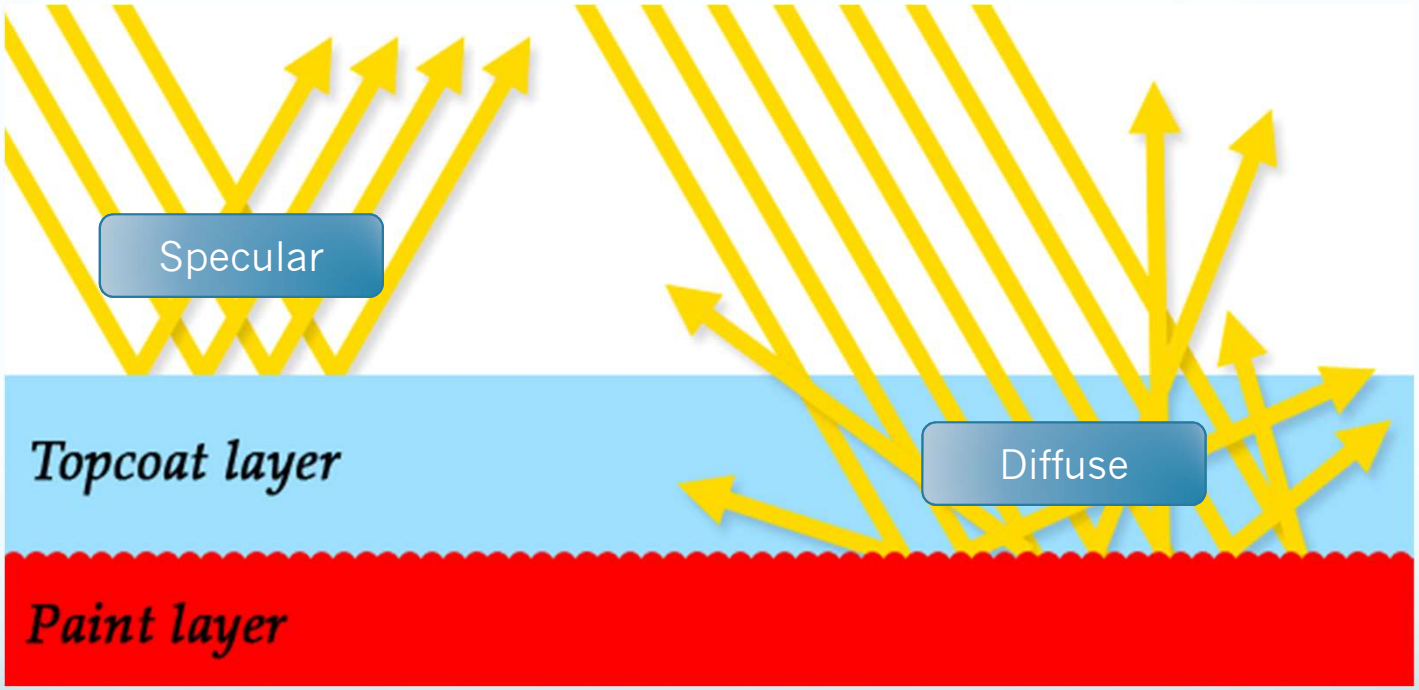


What about shiny surfaces with non-diffuse (specular) reflection?



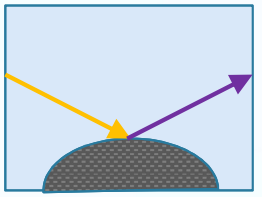
Specular and Diffuse Reflection



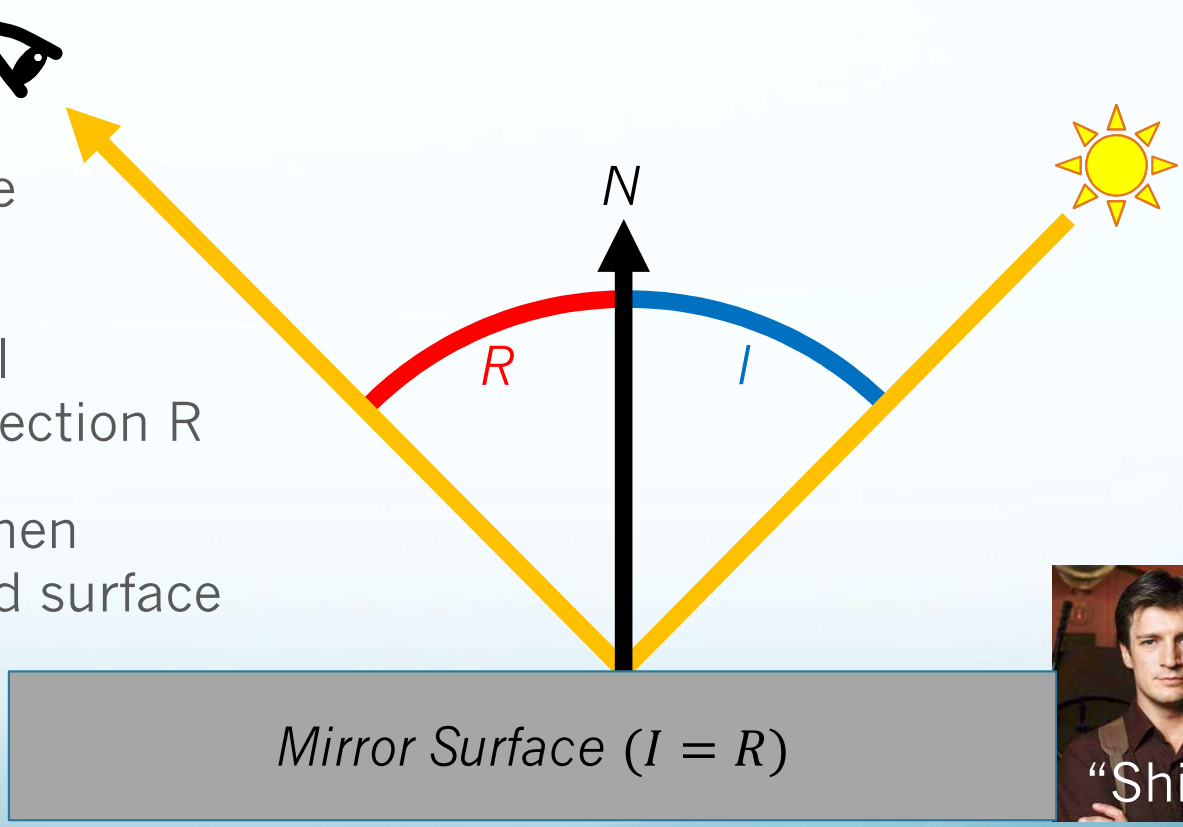


Credit: Tom Dalling

Specular Reflection

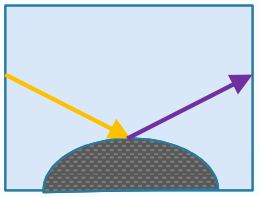


- Specular reflection also depends on the normal vector
- Angle of incidence I equals angle of reflection R
- Light visible only when source, camera, and surface normal
  align just right

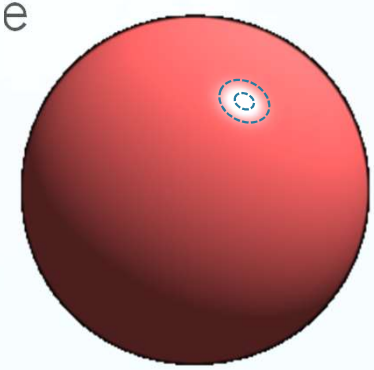


Credit: Tom Dalling

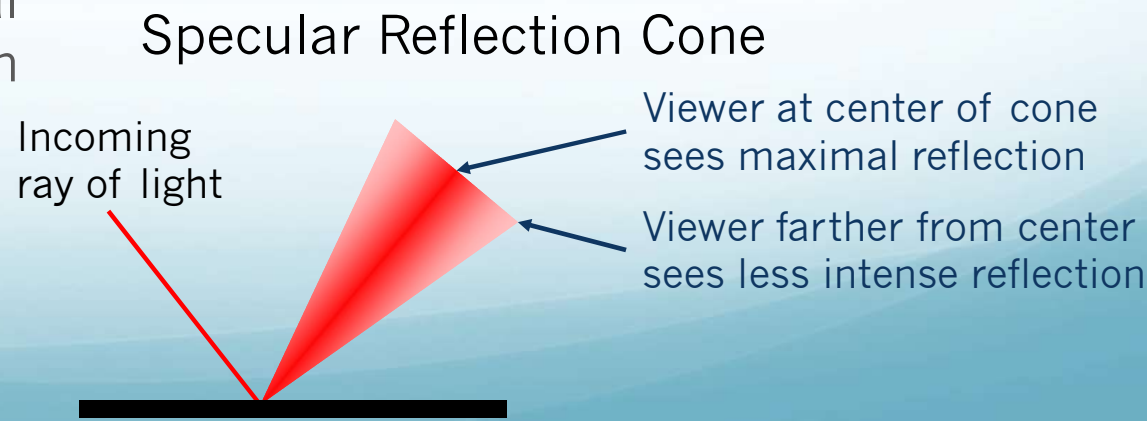
Specular Reflection



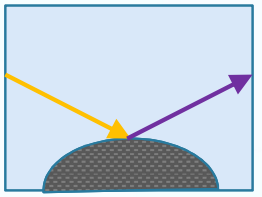
Why does specular reflection on a curved surface show a spot and not just a single point?



- Lights are usually not point sources
- Surfaces are not perfectly smooth at the microscopic level
- Deviations from perfect normal spread the reflection

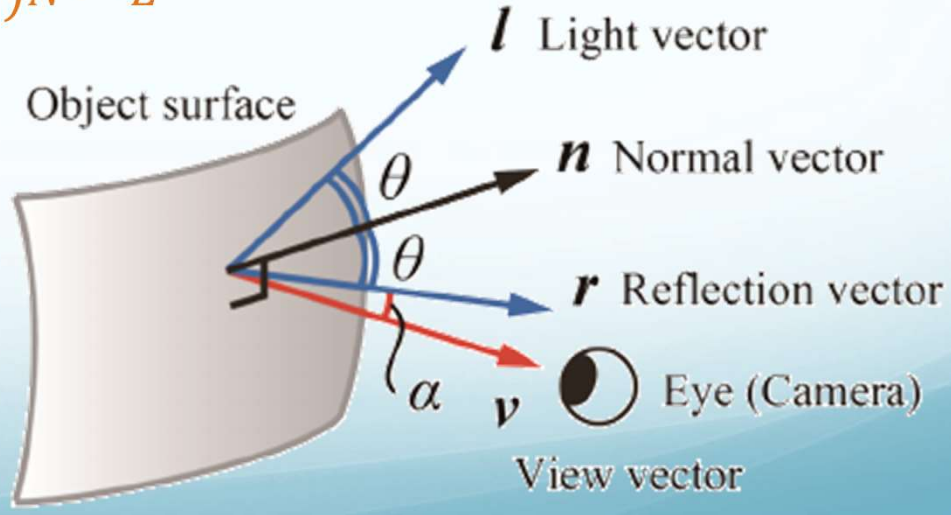


Phong Reflection Model



Phong (1975) proposed an empirical method to generate specular highlights on top of smooth shading

- Scant basis in physics but plausible & easy to compute
- Reflectance unit vector: $\hat{R} = 2(\hat{L} \cdot \hat{N})\hat{N} - \hat{L}$
- Specular highlight: $k_s(\hat{R} \cdot \hat{V})^{\alpha}$
- Diminishes quickly at angles away from $\hat{R}$ (increasing α)
- Parameter k_s is strength



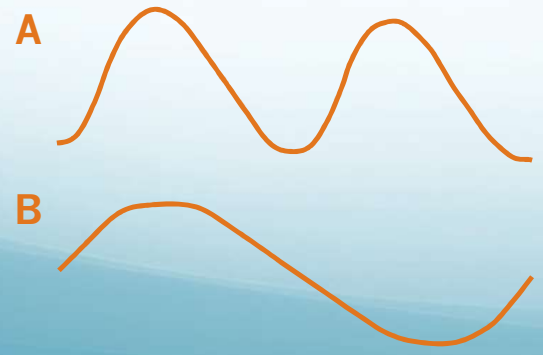
https://www.researchgate.net/figure/Light-reflection-on-an-object-surface-for-the-Phong-reflection-

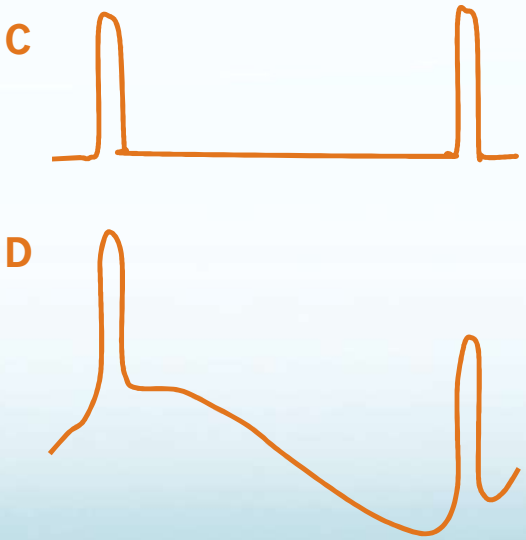
Questions

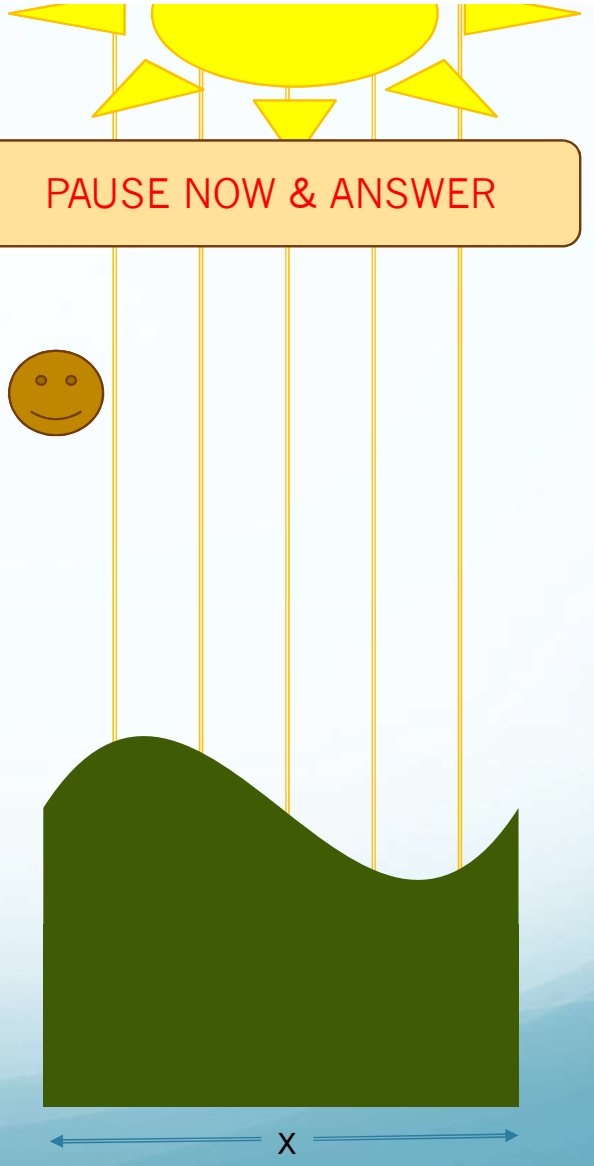
PAUSE NOW & ANSWER

Consider the picture at right. Assume vertical directional light and observer as shown. Match the descriptions below with the appropriate graph.

1. Diffuse reflectance **A**
2. Specular reflectance **C**
3. Combined reflectance **D**
4. None of the above **B**



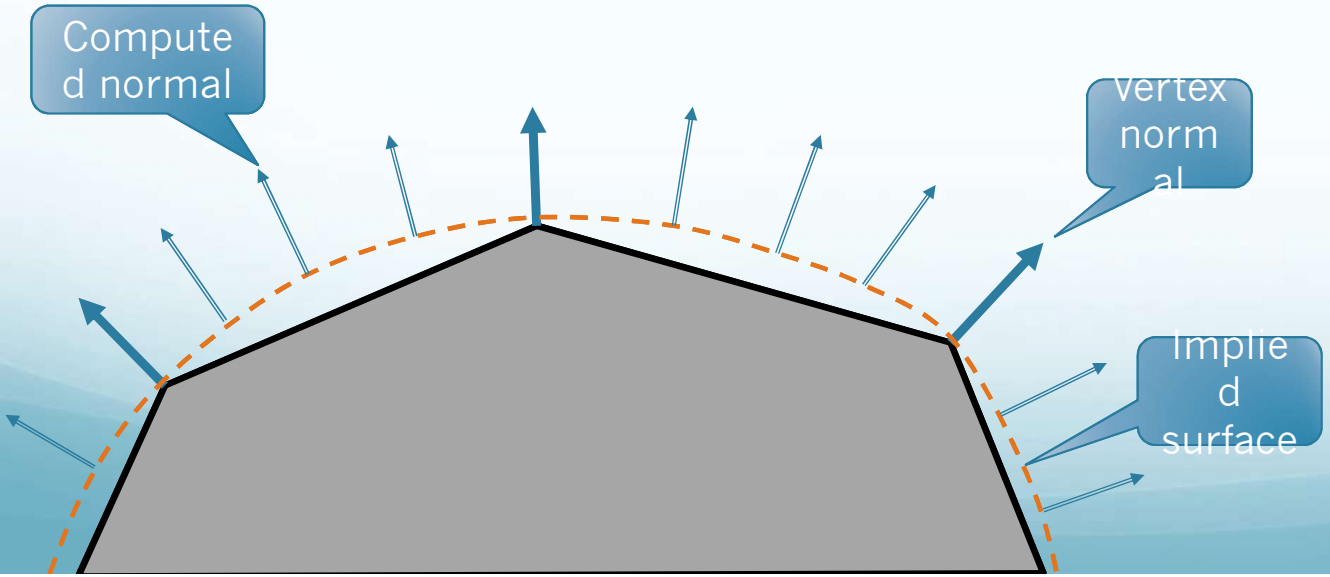


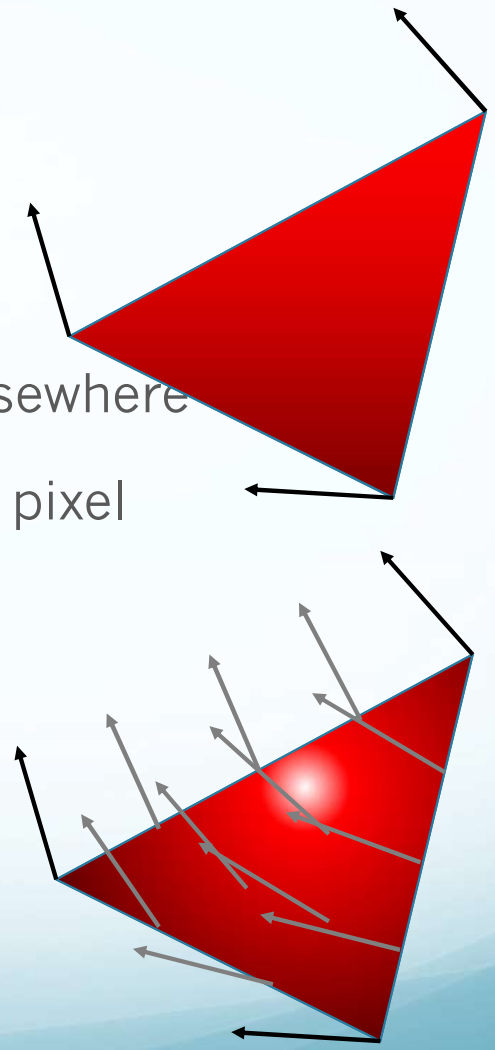


Phong Shading

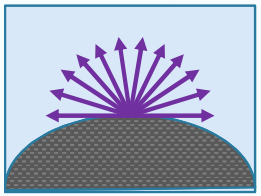
Phong requires a more complex shading computation

- Goraud: compute colors at 3 corners & interpolate elsewhere
- Phong: interpolate normals & compute color at each pixel



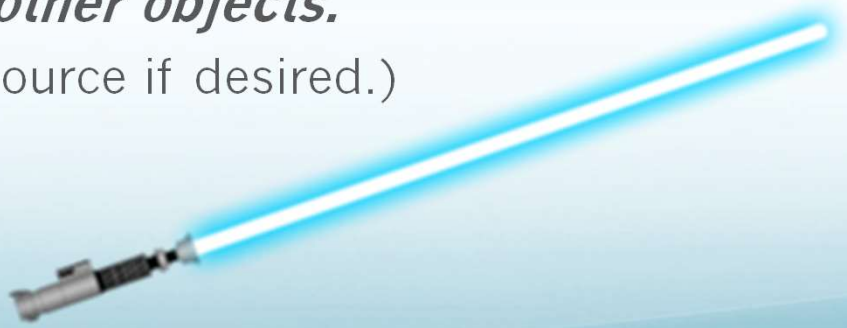


Emitted Light



Emitted light appears to come from the object surface without any external source. Like ambient light, intensity does not depend on 3D shape.

- Appears to glow internally
- Like ambient light, value is same everywhere so looks “flat”, not 3D
- ***Does not illuminate other objects.***
  (Combine with a light source if desired.)

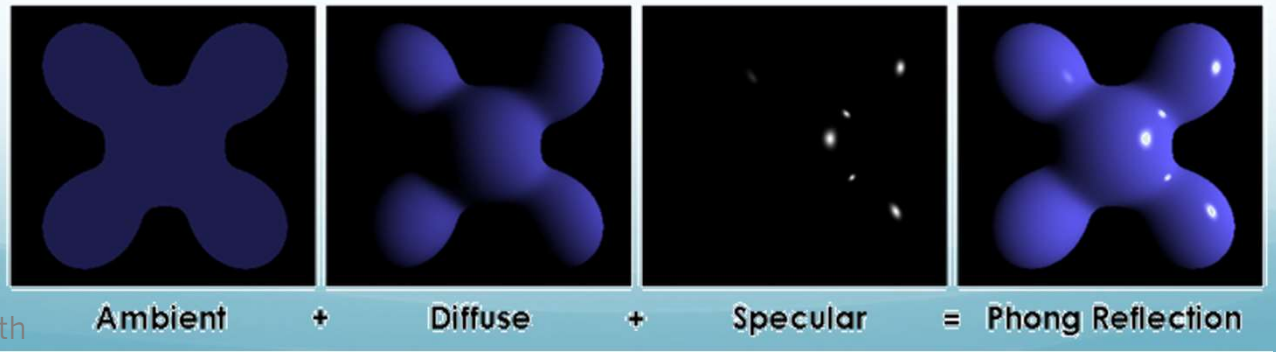


Combined Light Transmission

All sources of light add together by the Three.js pixel shader

Pixel color = (ambient) + (diffuse) + (specular) + (emitted)

- Each color component adds separately:
  $(\mathrm{R}, \mathrm{G}, \mathrm{B}) = (R_a + R_d + R_s + R_e, G_a + G_d + G_s + G_e, B_a + B_d + B_s + B_e)$
- Total can add to more than maximum color value (255)
  - Clipping occurs for the affected component in this case
  - Result: "washout" areas like an overexposed picture (white or other)

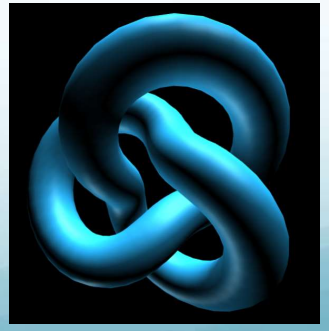


Credit:
Brad Smith

Lights & Shading in Three.js

Desired shading effects can be specified several ways in three.js:

- Many properties are specified by choosing the material of an object (e.g., `MeshPhongMaterial`, `MeshLambertMaterial`)
- Some have settable properties (color, transparency, reflectance parameter)
  - Look in three.js documentation
- Some behavior also specified via additional flags (e.g., `Material.flatShading`)



Questions

PAUSE NOW & ANSWER

1. Match the following (A-C to L-M to X-Z):
   - A. Goraud Shading — L. Three normal per face, interpolate normals
   - B. Phong Shading — M. Three normal per face, interpolate colors
   - C. Flat Shading — N. Single normal per face, no interpolation

2. Which color component(s) are washed out?
   - Ambient RGB: (128,64,0)
   - Diffuse RGB: (32, 200, 0)
   - Specular RGB: (128,10,0)
   - Emissive RGB: (0,0,255)

Sum = (288,274,255)
Red and Green are >255

3. What will be the final color when the above are combined?

White

Review

After watching this video you should be able to...

- Define a BRDF and explain how it is measured
- List and describe the four types of light transmission in Three.js
- Compute how geometry interacts with light position in diffuse shading
- Describe and apply three shading algorithms, along with their different schemes for handling surface normal
- Understand how Three.js computes the final color of a pixel
- Write Three.js programs that control shading to achieve desired effects

Music: https://www.bensound.com