

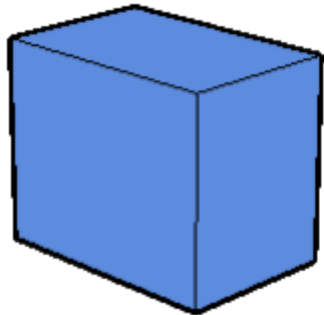
# EPL 426

## Lab 3 - Phong Shading

Andreas Andreou

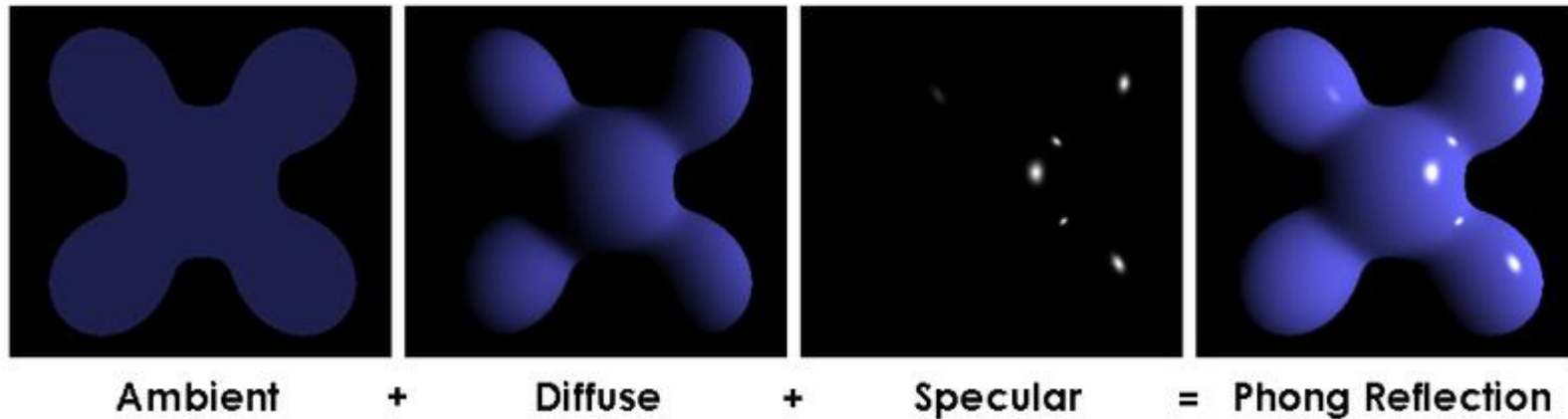
# Phong Shading

- ▶ Ένα από τα βασικά μοντέλα τοπικού φωτισμού είναι το Phong Shading
- ▶ Το μοντέλο αυτό ερμηνεύει ουσιαστικά τον τρόπο σκίασης των αντικειμένων στο χώρο με βάση το φωτισμό που υπάρχει.



# Phong Shading

- ▶ Το Phong Shading αποτελείται από τρεις βασικές συνιστώσες:
  - ▶ Έμμεσο φωτισμό(ambient light)
  - ▶ Διάχυτη ανάκλαση φωτός (diffuse reflection)
  - ▶ Κατευθυνόμενη ανάκλαση φωτός (specular reflection)



# Ambient Light

- ▶ Ο έμμεσος φωτισμός είναι μια προσέγγιση του γενικού φωτισμού που καθορίζει πώς ένα αντικείμενο φωτίζεται από αδέσποτο φως
- ▶ Είναι σταθερός σε όλη την επιφάνεια του αντικειμένου
- ▶ Συχνά χρησιμοποιείται για να σιγουρευτούμε ότι όλα φωτίζονται
- ▶ Αν προσέξετε καλά στο Lambert.cpp (Miro) ήδη υπάρχει ambient φωτισμός σε αυτήν την περίπτωση

```
// add the ambient component  
L += m_ka;
```

# Ambient Light

$$I_r = k_a I_a$$

**$k_a$**  = is an ambient reflection constant, the ratio of reflection of the ambient term present in all points in the scene rendered (0-1 -> float/double)

**$I_a$**  = controls the ambient lighting (RGB -> Vector 3)

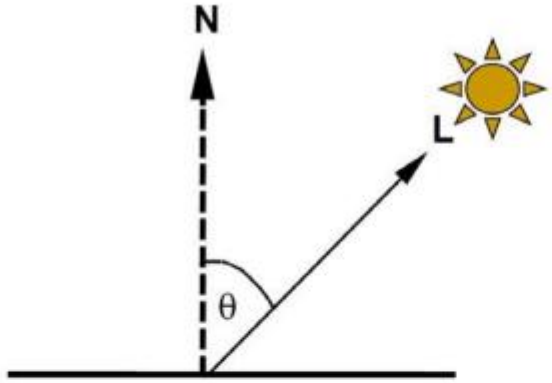
# Diffuse Reflection

- ▶ Στη διάχυτη ανάκλαση (diffuse reflection) το φως ανακλάται εξίσου προς όλες τις κατευθύνσεις
- ▶ Αυτό γίνεται σύμφωνα με το νόμο του Lambert
- ▶ Αν κοιτάξετε στο solution του Miro που σας έχουμε δώσει, έχει υλοποιημένο το diffuse reflection με το νόμο του Lambert

```
// get the diffuse component  
float nDotL = dot(hit.N, l);  
Vector3 result = pLight->color();  
result *= m_kd;
```

- ▶ Το ποσοστό του φωτός που ανακλάται διάχυτα εξαρτάται από:
  - ▶ Συντελεστή διάχυτης ανάκλασης (kd)
  - ▶ Την κατεύθυνση και ένταση του προσπίπτοντος φωτός

# Diffuse Reflection



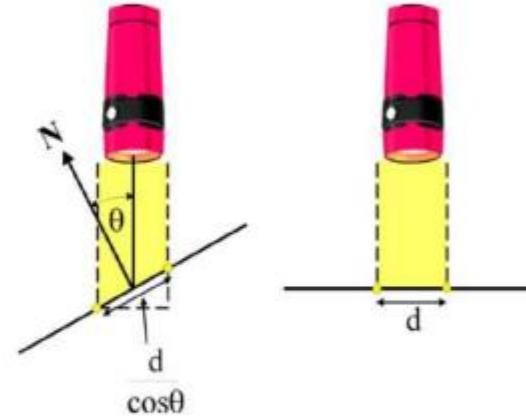
Ανάκλαση είναι ανάλογη του  $\cos \Theta$

$$I_r = I k_d \cos \Theta$$

**L** είναι η κατεύθυνση της φωτεινής πηγής

**N** είναι η κάθετος

**I** είναι η ένταση της φωτεινής πηγής

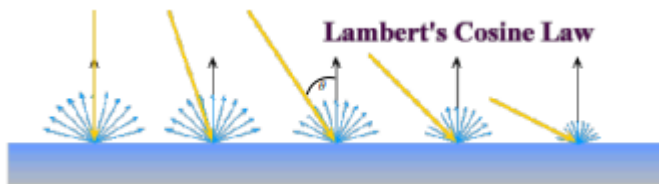


## Trigonometry Table

	0°	30°	45°	60°	90°
sin $\theta$	0	$\frac{1}{2}$	$\frac{1}{\sqrt{2}}$	$\frac{\sqrt{3}}{2}$	1
cos $\theta$	1	$\frac{\sqrt{3}}{2}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{2}$	0
tan $\theta$	0	$\frac{1}{\sqrt{3}}$	1	$\sqrt{3}$	Not defined
cosec $\theta$	Not defined	2	$\sqrt{2}$	$\frac{2}{\sqrt{3}}$	1
sec $\theta$	1	$\frac{2}{\sqrt{3}}$	$\sqrt{2}$	2	Not defined
cot $\theta$	Not defined	$\sqrt{3}$	1	$\frac{1}{\sqrt{3}}$	0

# Diffuse Reflection

- ▶ Κοιτάξετε στο Miro solution το αρχείο Lambert.cpp
- ▶ Συσχετίστε τον πιο πάνω κανόνα του Lambert με την υλοποίηση στο αρχείο αυτό
- ▶ Το μόνο επιπρόσθετο στοιχείο στην υλοποίηση είναι η χρήση του Inverse Square Law of light που ουσιαστικά συσχετίζει την απόδοση του φωτός σχετικά με την απόσταση εκπομπής του



```
// loop over all of the lights
Lights::const_iterator lightIter;
for (lightIter = lightlist->begin(); lightIter != lightlist->end(); lightIter++)
{
    PointLight* pLight = *lightIter;

    Vector3 l = pLight->position() - hit.P;

    // the inverse-squared falloff
    float falloff = 1./l.length2();

    // normalize the light direction
    l /= sqrt(falloff);

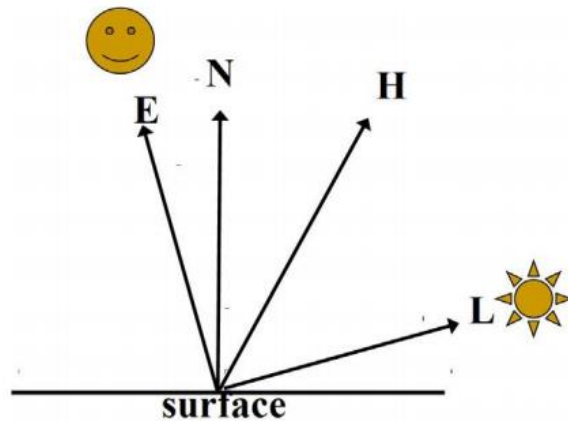
    // get the diffuse component
    float nDotL = dot(hit.N, l);
    Vector3 result = pLight->color();
    result *= m_kd;

    L += std::max(0.0f, nDotL/falloff * pLight->wattage() / PI) * result;
}
```

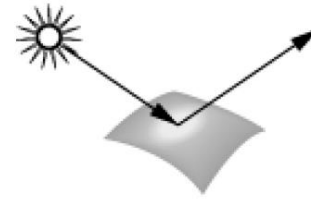


# Specular Reflection

- ▶ Σε αυτή την περίπτωση έχουμε κατευθυνόμενη ανάκλαση η οποία μας οδηγεί στο αποτέλεσμα σχηματισμού highlights

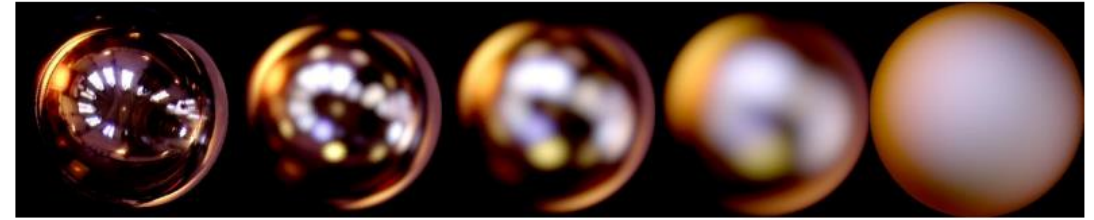


- **E** είναι η κατεύθυνση προς το μάτι
- **N** είναι η κάθετος
- **L** είναι η κατεύθυνση προς την φωτεινή πηγή
- Η **H** είναι η διχοτόμος των **E** και **L**



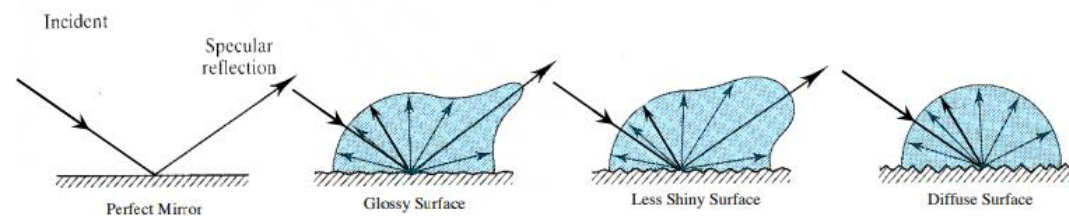
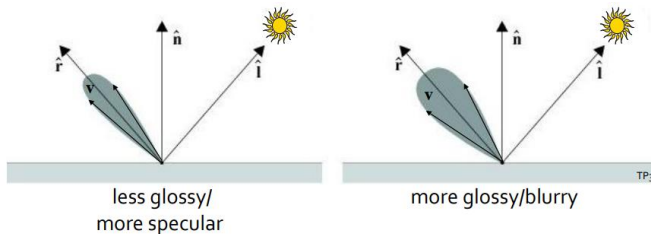
# Specular Reflection

increasing roughness



$$k_s I (\mathbf{h} \cdot \mathbf{n})^m$$

- ▶ Το  $m$  καθορίζει τα χαρακτηριστικά του highlight
  - ▶ Μεγάλο  $m$  συνεπάγεται highlight συγκεντρωμένο γυρο απο την ανακλώμενη
  - ▶ Μικρό  $m$  κάνει το highlight πιο πλατύ και θολό



# Phong Shading

For each light

$$I_r = k_a I_a + \sum_{j=1}^p I_j \left( k_d (\mathbf{n} \cdot \mathbf{l}_j) + k_s (\mathbf{h}_j \cdot \mathbf{n})^m \right)$$

Ambient Light

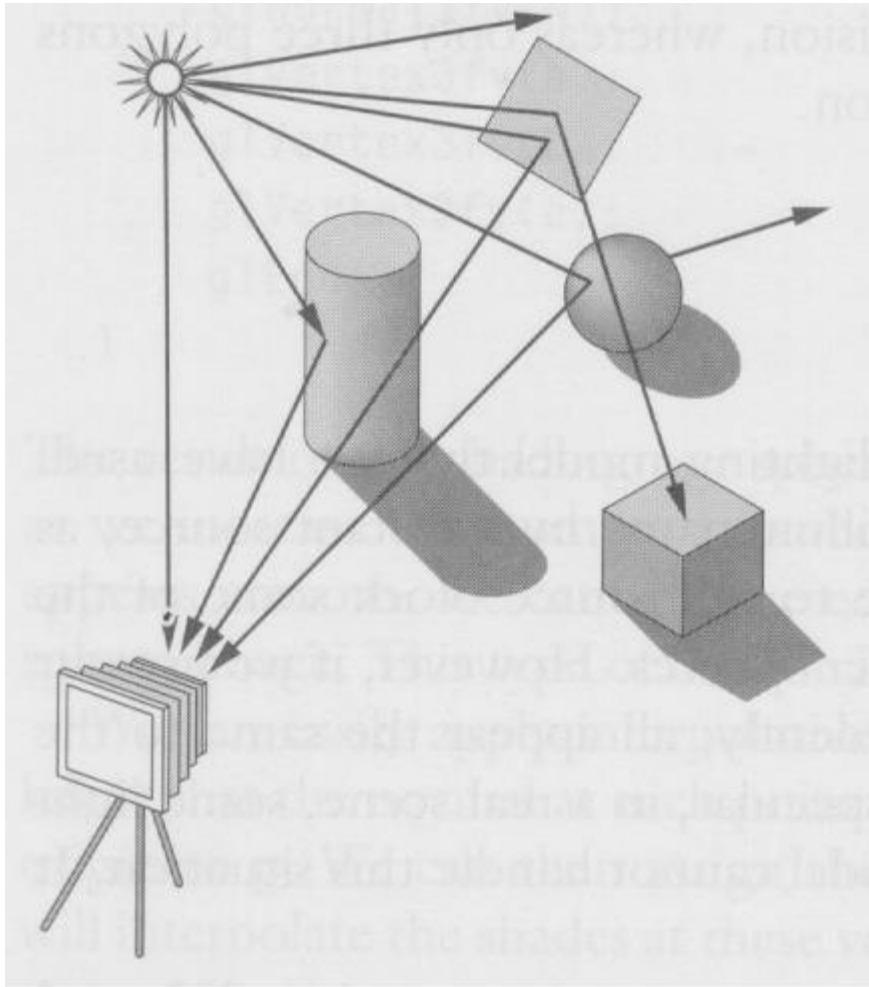
Diffuse Reflection

Specular Reflection

# Phong Shading

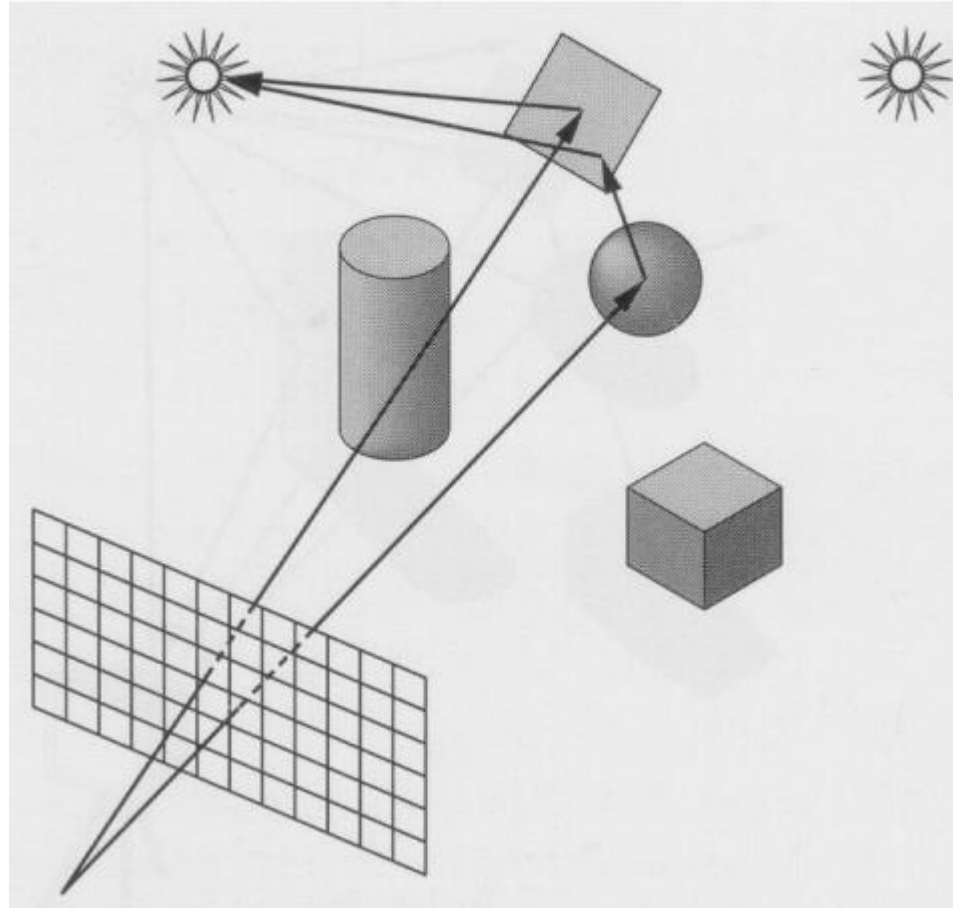


<http://www.cs.toronto.edu/~jacobson/phong-demo/>



Forward ray tracing  
(and real world)

### Backward ray tracing



# Test scene - ray traced

