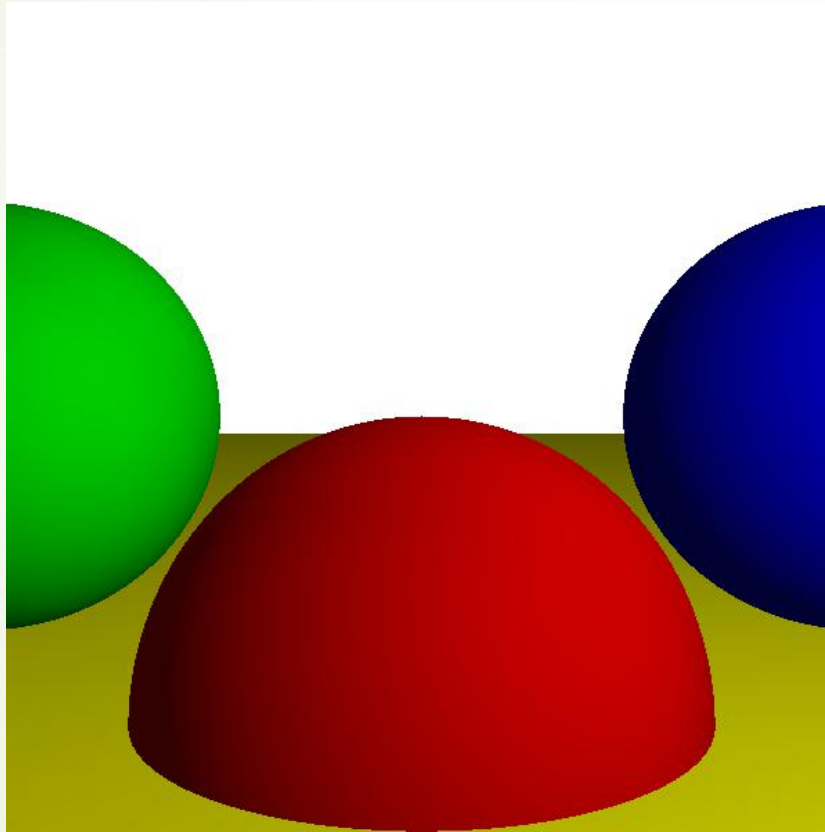


Γραφικά Υπολογιστών

Μάθημα 5 - Φως, μέρος II

Γιώργος Σφήκας

Φως



- ❏ Στο προηγούμενο μάθημα μιλήσαμε για τρεις πηγές φωτός και δύο είδη επιφανειών
- ❏ Είδαμε πως υλοποιούμε την διάχυτη ανάκλαση (diffuse reflection), που αντιστοιχεί σε «matte» επιφάνειες
- ❏ Σήμερα θα μιλήσουμε για μη-διάχυτη ανάκλαση: Το φως δεν διαχέεται ισομερώς προς όλες τις κατευθύνσεις

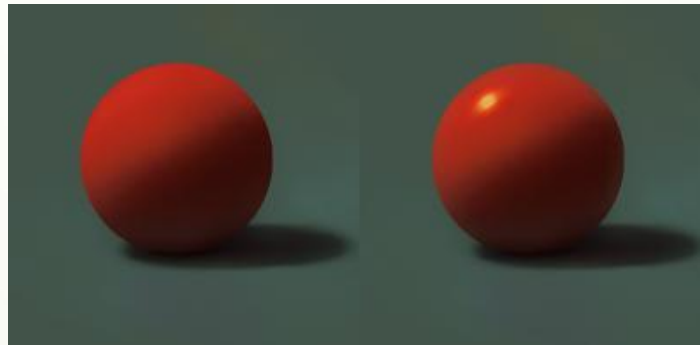


Η εξίσωση που χρειαστήκαμε για την
“διάχυτη ανάκλαση”..

$$I_P = I_A + \sum_{i=1}^n I_i \frac{\langle \vec{N}, \vec{L}_i \rangle}{|\vec{N}| |\vec{L}_i|}$$

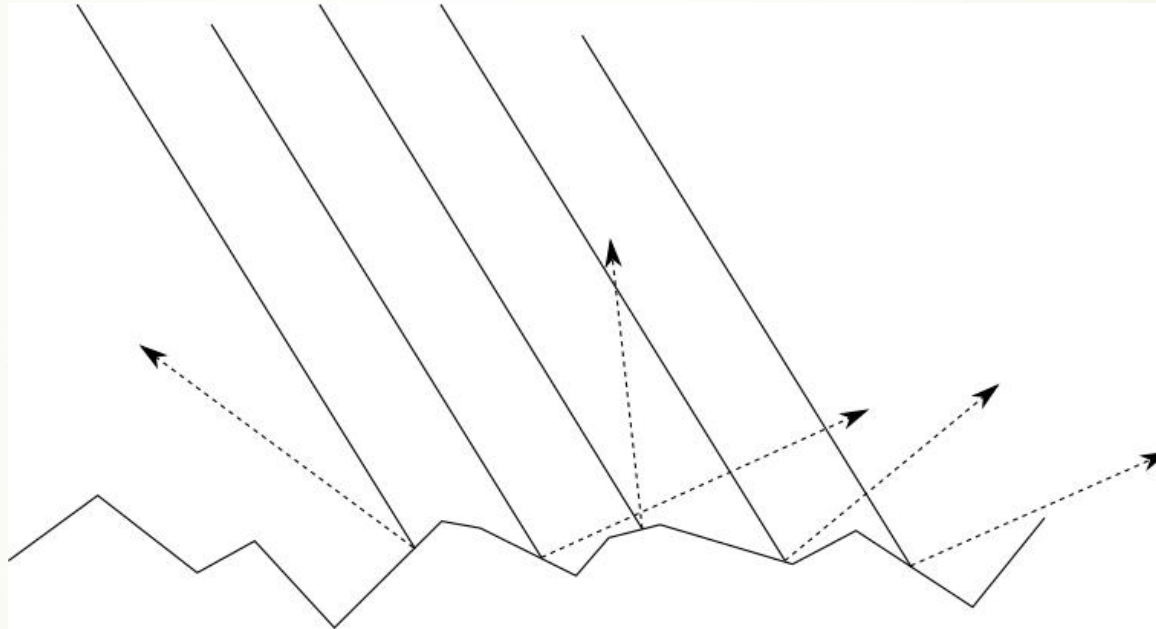
Κατοπτρική ανάκλαση (*Specular reflection*)

- ❏ Μας ενδιαφέρουν τα “γυαλιστερά” αντικείμενα
- ❏ = το πως φαίνονται εξαρτάται από που τα κοιτάζουμε (αντίθετα στα *matte*, μας ενδιέφεραν μόνο τα διανύσματα θέσης για το «*P*» και το κανονικό διάνυσμα «*N*»)
- ❏ Φωτεινά σημεία που φαίνεται να “μας ακολουθούν”
- ❏ Γιατί συμβαίνει αυτό ;



Κατοπτρική ανάκλαση (*Specular reflection*)

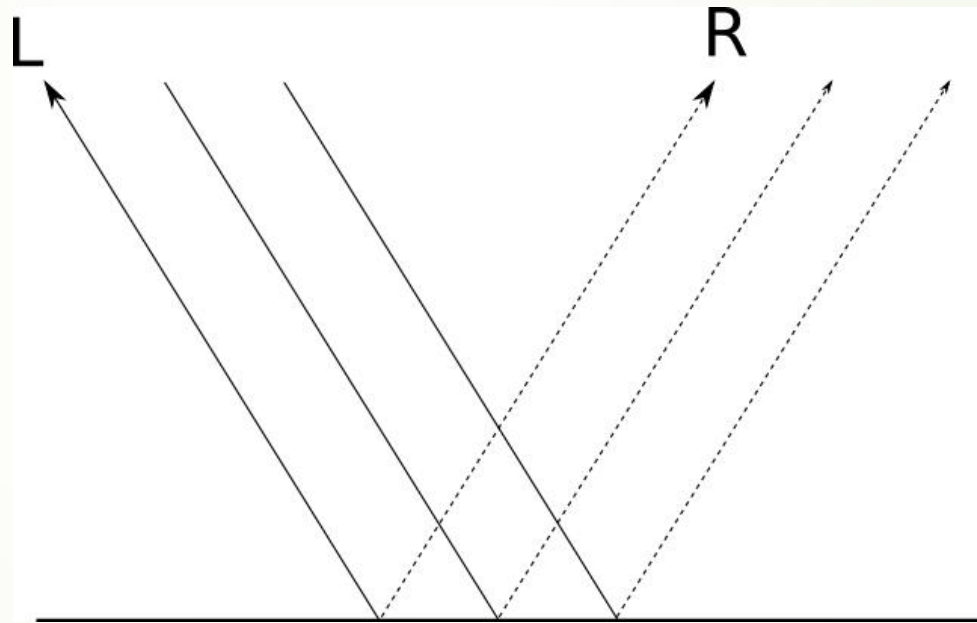
Λίγη ακόμα Φυσική



Η τραχειά επιφάνεια ενός “ματ” αντικειμένου, όπως θα φαινόταν από ένα μικροσκόπιο. Οι προσπίπτουσες ακτίνες ανακλώνται σε τυχαίες διευθύνσεις.

Κατοπτρική ανάκλαση (*Specular reflection*)

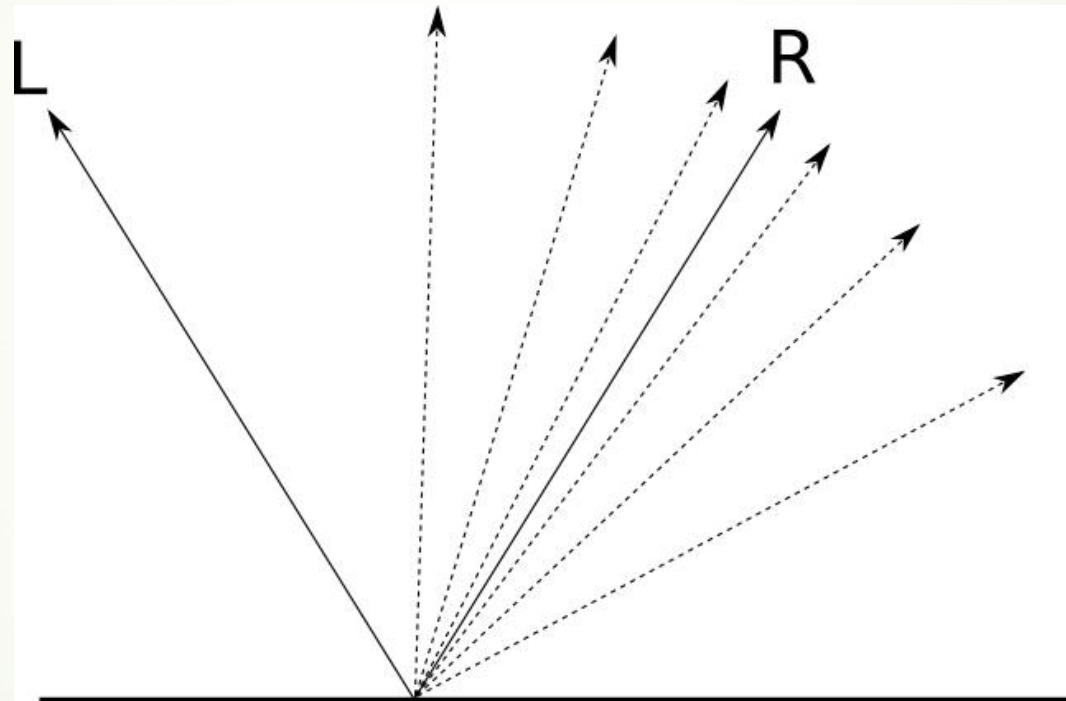
Λίγη ακόμα Φυσική



Ακτίνες που προσπίπτουν και ανακλώνται σε έναν καθρέφτη

Κατοπτρική ανάκλαση (*Specular reflection*)

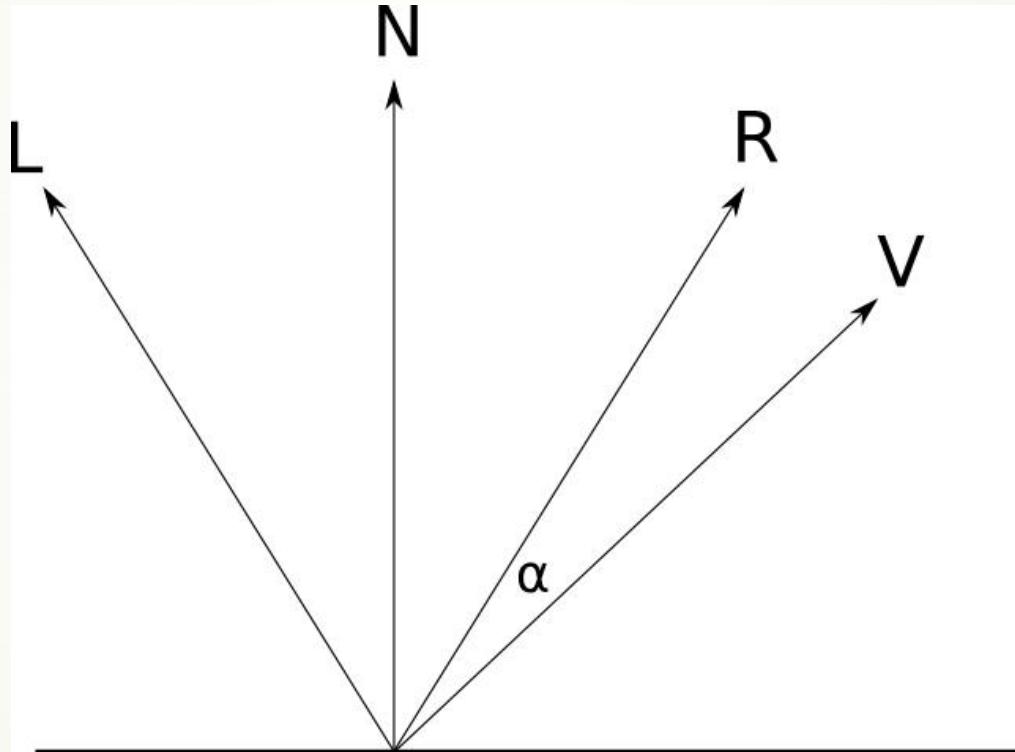
Λίγη ακόμα Φυσική



Στο βαθμό που μια επιφάνεια είναι τέλειος καθρέφτης, οι κατευθύνσεις ανάκλασης συγκεντρώνονται γύρω από την κατεύθυνση R

Κατοπτρική ανάκλαση (*Specular reflection*)

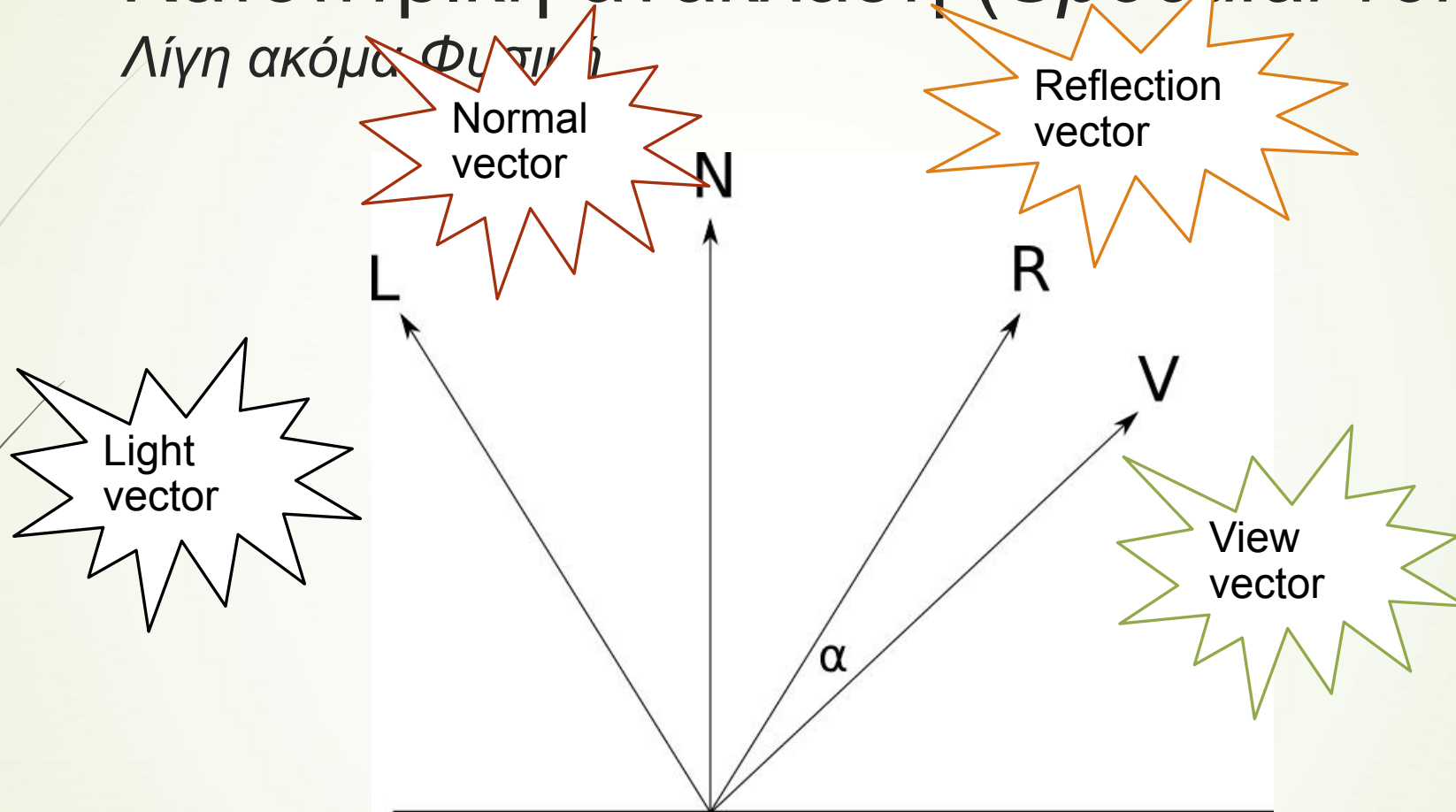
Λίγη ακόμα Φυσική



Τα διανύσματα και οι γωνίες που χρειαζόμαστε για τον υπολογισμό της κατοπτρικής ανάκλασης

Κατοπτρική ανάκλαση (*Specular reflection*)

Λίγη ακόμα Φυσική

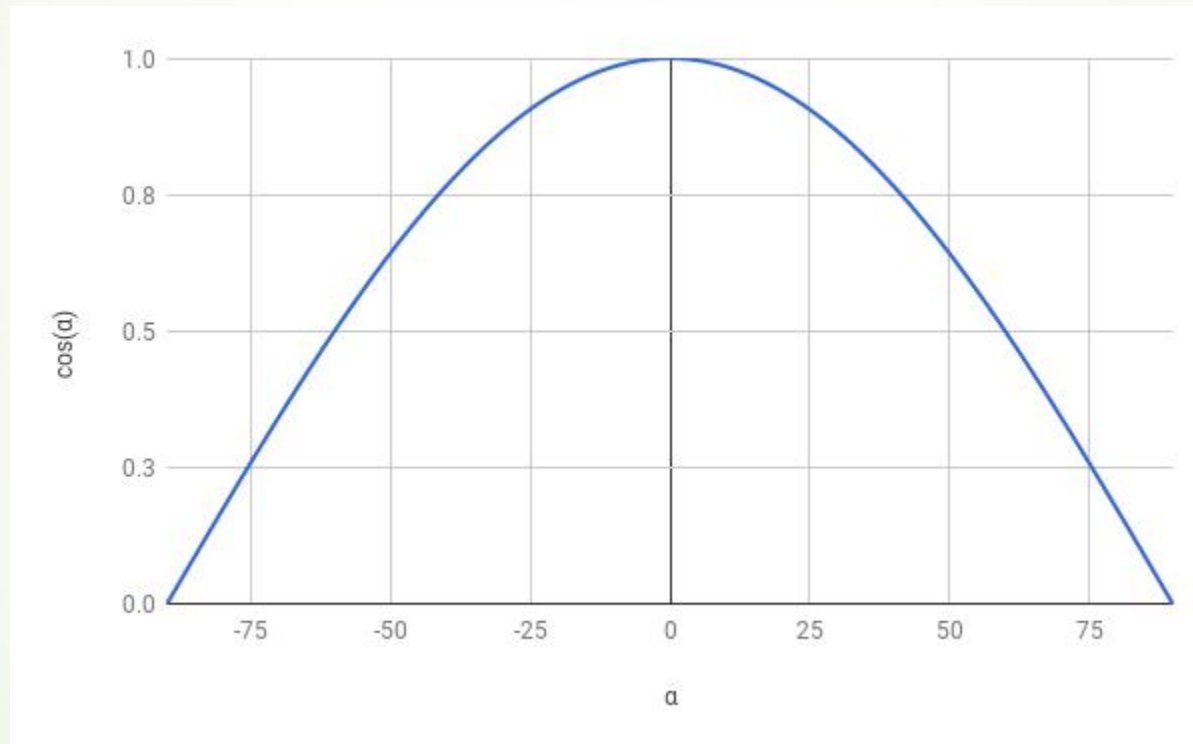


Τα διανύσματα και οι γωνίες που χρειαζόμαστε για τον υπολογισμό της κατοπτρικής ανάκλασης

Κατοπτρική ανάκλαση (*Specular reflection*)

Απλουστευτικές υποθέσεις

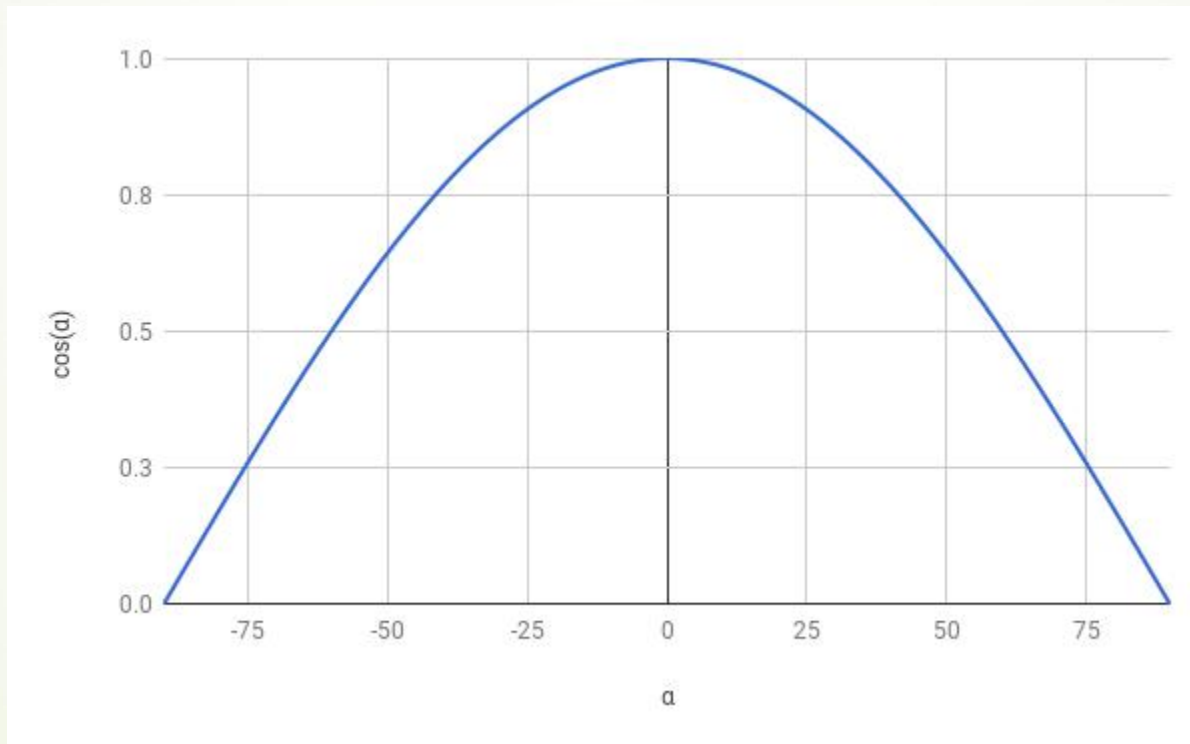
☞ Θέλουμε πάλι το συνημίτονο μιας γωνίας ... ή όχι;



Κατοπτρική ανάκλαση (*Specular reflection*)

Απλουστευτικές υποθέσεις

📏 Θέλουμε πάλι το συνημίτονο μιας γωνίας ... ή όχι;



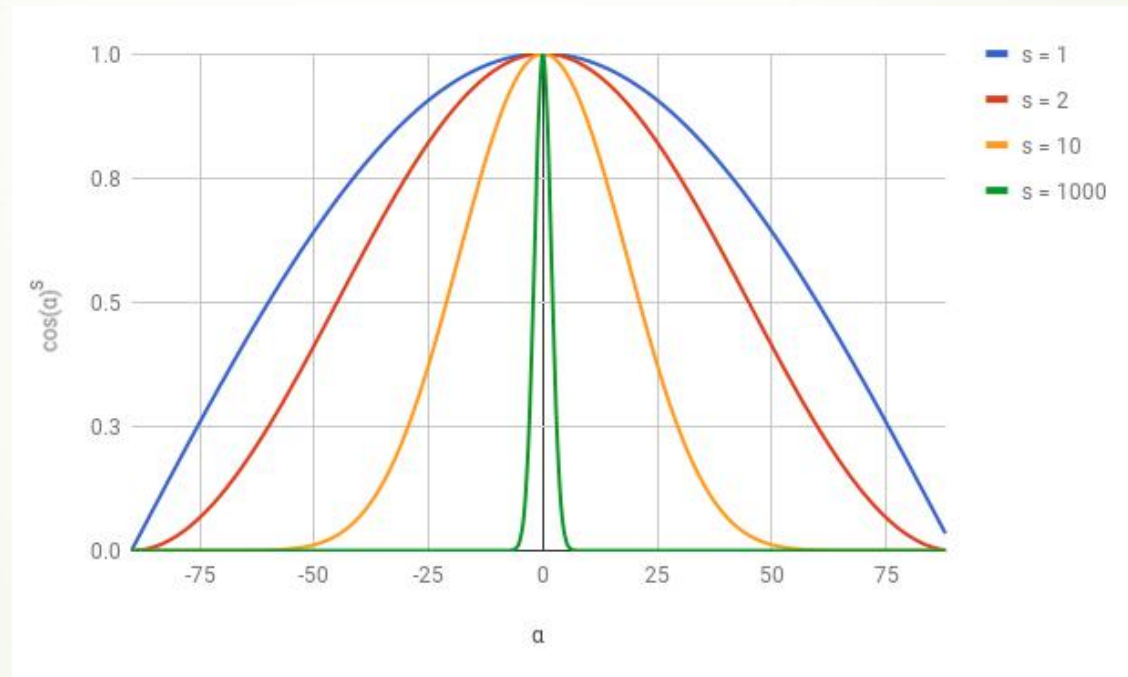
Πρόβλημα: Όλα
τα αντικείμενα
θα είναι το ίδιο
λαμπερά..

Κατοπτρική ανάκλαση (*Specular reflection*)

Απλουστευτικές υποθέσεις

- Ακόμα μια χρήσιμη μοντελοποίηση (= διαζύγιο από τη Φυσική)
- Αφού $0 \leq \cos(a) \leq 1$, τότε και $0 \leq \cos(a)^s \leq 1$

Μεγαλύτερο s =
πιο “μαζεμένο”
γράφημα

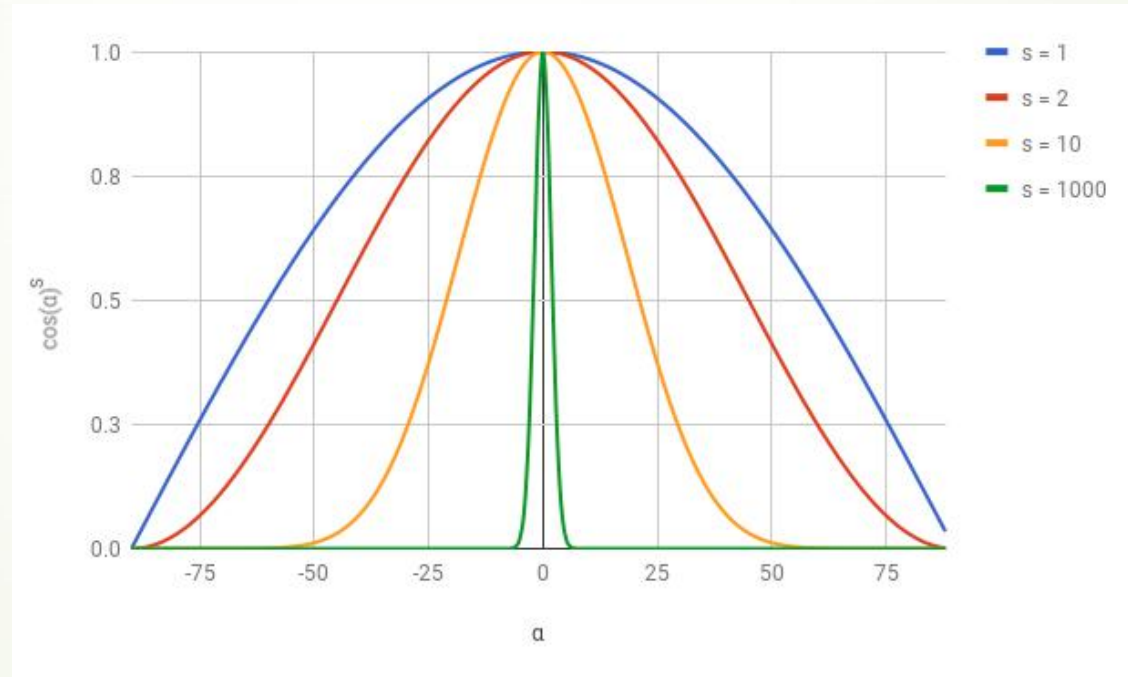


Κατοπτρική ανάκλαση (*Specular reflection*)

Απλουστευτικές υποθέσεις

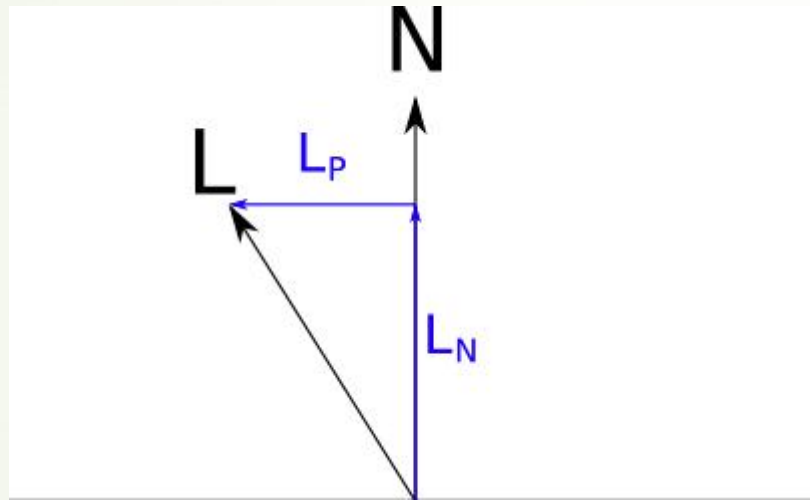
- Ακόμα μια χρήσιμη μοντελοποίηση (= διαζύγιο από τη Φυσική)
- Αφού $0 \leq \cos(\alpha) \leq 1$, τότε και $0 \leq \cos(\alpha)^s \leq 1$

Μεγαλύτερο s =
πιο “μαζεμένο”
γράφημα

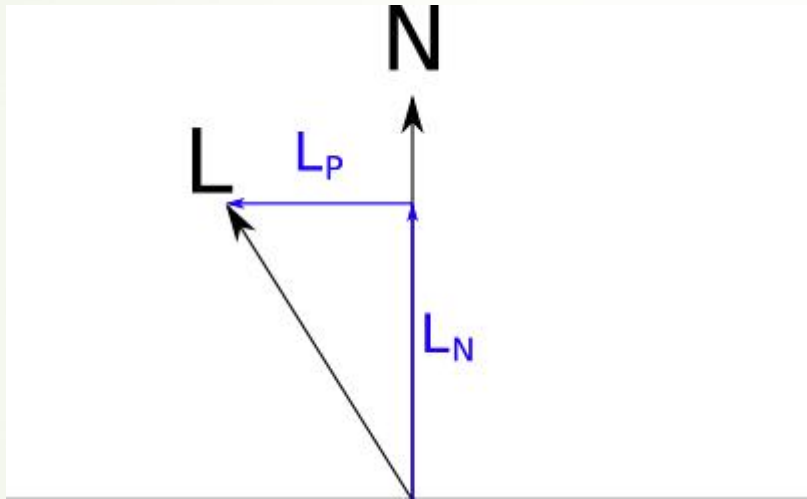


Ιδέα : Θα
μοντελοποιήσουμε τα
πιο “κατοπτρικά”
αντικείμενα με
μεγαλύτερο s !

Υπολογισμός R και V

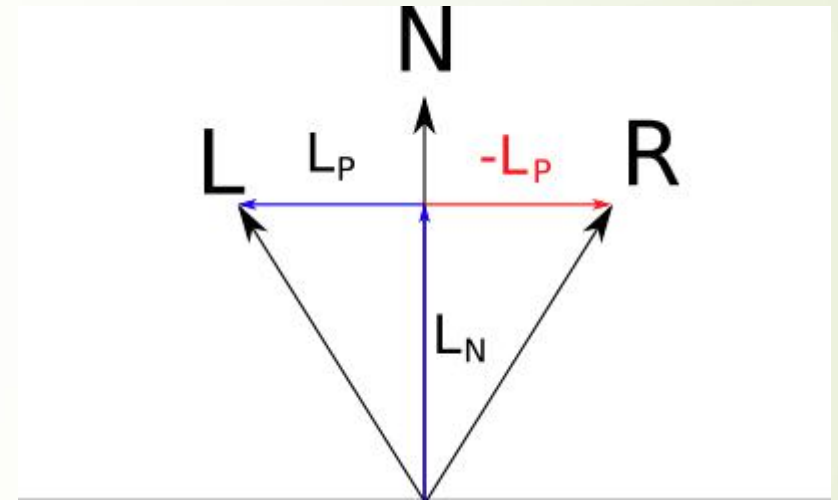
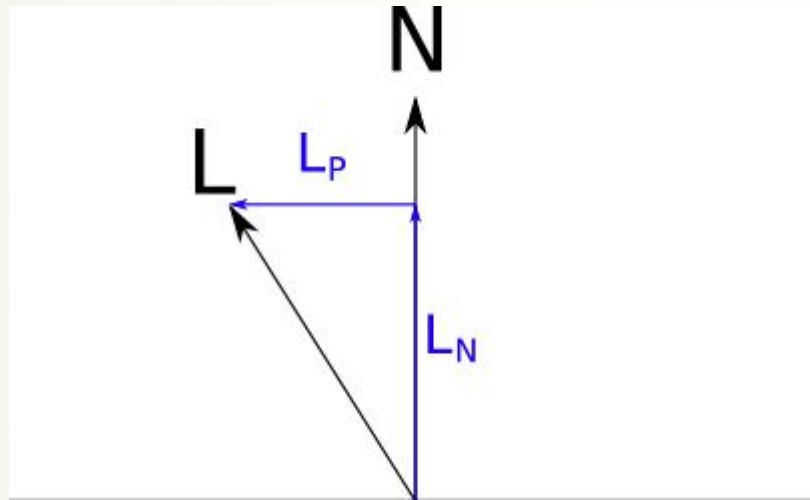


Υπολογισμός R και V



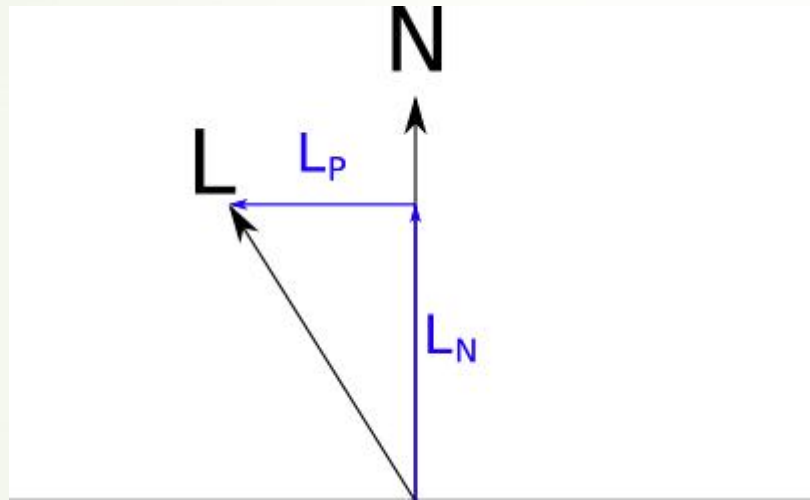
$$\vec{L}_p = \vec{L} - \vec{L}_N = \vec{L} - \vec{N}\langle\vec{N}, \vec{L}\rangle$$

Υπολογισμός R και V

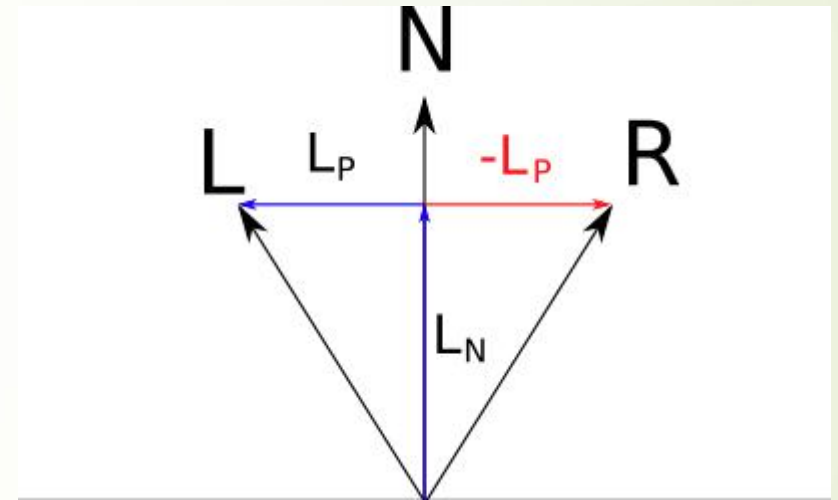


$$\vec{L}_p = \vec{L} - \vec{L}_N = \vec{L} - \vec{N}\langle\vec{N}, \vec{L}\rangle$$

Υπολογισμός R και V

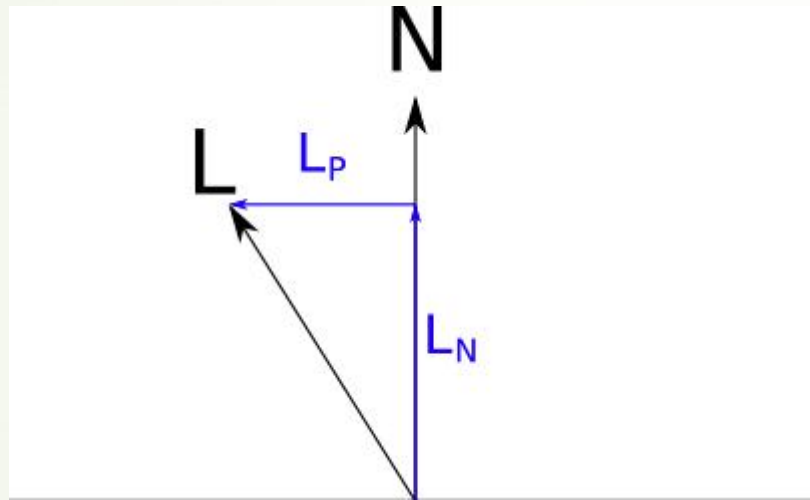


$$\vec{L}_p = \vec{L} - \vec{L}_N = \vec{L} - \vec{N}\langle\vec{N}, \vec{L}\rangle$$

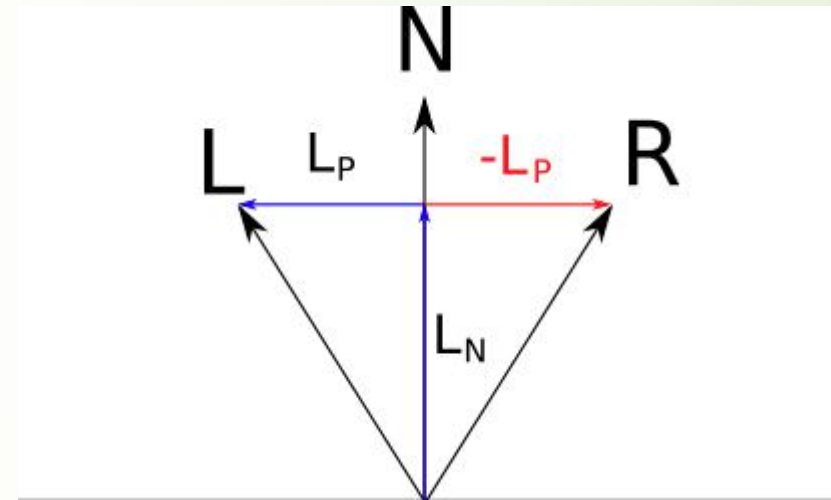


$$\vec{R} = \vec{N}\langle\vec{N}, \vec{L}\rangle - \vec{L} + \vec{N}\langle\vec{N}, \vec{L}\rangle$$

Υπολογισμός R και V



$$\vec{L}_p = \vec{L} - \vec{L}_N = \vec{L} - \vec{N}\langle\vec{N}, \vec{L}\rangle$$



$$\vec{R} = 2\vec{N}\langle\vec{N}, \vec{L}\rangle - \vec{L}$$

Υπολογισμός του όρου για την κατοπτρική ανάκλαση

$$\vec{R} = 2\vec{N}\langle\vec{N}, \vec{L}\rangle - \vec{L}$$
$$I_S = I_L \left(\frac{\langle\vec{R}, \vec{V}\rangle}{|\vec{R}||\vec{V}|} \right)^s$$

- ❏ Όπως για τη διάχυτη ανάκλαση, πρέπει να προσέξουμε κάποιες περιπτώσεις..
 - ❏ Μηδενίζουμε τον όρο όταν $\cos(a)$ αρνητικό
 - ❏ Μηδενίζουμε τον όρο όταν το αντικείμενο είναι matte

Η ολοκληρωμένη εξίσωση

..για διάχυτη + κατοπτρική ανάκλαση

$$I_P = I_A + \sum_{i=1}^n I_i \cdot \left[\frac{\langle \vec{N}, \vec{L}_i \rangle}{|\vec{N}| |\vec{L}_i|} + \left(\frac{\langle \vec{R}_i, \vec{V} \rangle}{|\vec{R}_i| |\vec{V}|} \right)^s \right]$$

Υλοποίηση

```
sphere {  
  center = (0, -1, 3)  
  radius = 1  
  color = (255, 0, 0) # Red  
  specular = 500 # Shiny  
}  
sphere {  
  center = (2, 0, 4)  
  radius = 1  
  color = (0, 0, 255) # Blue  
  specular = 500 # Shiny  
}  
sphere {  
  center = (-2, 0, 4)  
  radius = 1  
  color = (0, 255, 0) # Green  
  specular = 10 # Somewhat shiny  
}  
sphere {  
  center = (0, -5001, 0)  
  radius = 5000  
  color = (255, 255, 0) # Yellow  
  specular = 1000 # Very shiny  
}
```

Υλοποίηση

```
ComputeLighting(P, N, V, s) {
  i = 0.0
  for light in scene.Lights {
    if light.type == ambient {
      i += light.intensity
    } else {
      if light.type == point {
        L = light.position - P
      } else {
        L = light.direction
      }

      // Diffuse
      n_dot_l = dot(N, L)
      if n_dot_l > 0 {
        i += light.intensity * n_dot_l / (length(N) * length(L))
      }

      // Specular
      if s != -1 {
        R = 2 * N * dot(N, L) - L
        r_dot_v = dot(R, V)
        if r_dot_v > 0 {
          i += light.intensity * pow(r_dot_v / (length(R) * length(V)), s)
        }
      }
    }
  }
  return i
}
```

Υλοποίηση

```
TraceRay(O, D, t_min, t_max) {
    closest_t = inf
    closest_sphere = NULL
    for sphere in scene.Spheres {
        t1, t2 = IntersectRaySphere(O, D, sphere)
        if t1 in [t_min, t_max] and t1 < closest_t {
            closest_t = t1
            closest_sphere = sphere
        }
        if t2 in [t_min, t_max] and t2 < closest_t {
            closest_t = t2
            closest_sphere = sphere
        }
    }
    if closest_sphere == NULL {
        return BACKGROUND_COLOR
    }

    P = O + closest_t * D // Compute intersection
    N = P - closest_sphere.center // Compute sphere normal at intersection
    N = N / length(N)
    return closest_sphere.color * ComputeLighting(P, N, -D, closest_sphere.specular)
}
```

📏 Πώς υπολογίζω το View vector? Είναι απλά $=-D$ (αντίθετο από την ακτίνα παρακολούθησης)

Υλοποίηση

```
TraceRay(O, D, t_min, t_max) {
    closest_t = inf
    closest_sphere = NULL
    for sphere in scene.Spheres {
        t1, t2 = IntersectRaySphere(O, D, sphere)
        if t1 in [t_min, t_max] and t1 < closest_t {
            closest_t = t1
            closest_sphere = sphere
        }
        if t2 in [t_min, t_max] and t2 < closest_t {
            closest_t = t2
            closest_sphere = sphere
        }
    }
    if closest_sphere == NULL {
        return BACKGROUND_COLOR
    }

    P = O + closest_t * D // Compute intersection
    N = P - closest_sphere.center // Compute sphere normal at intersection
    N = N / length(N)
    return closest_sphere.color * ComputeLighting(P, N, -D, closest_sphere.specular)
}
```

Προσοχή -- κάποιες
χρωματικές τιμές μπορεί
να βγούν εκτός [0,255]

📏 Πώς υπολογίζω το View vector? Είναι απλά $=-D$ (αντίθετο από την ακτίνα παρακολούθησης)



Στο εργαστήριο..

- ❏ Θα υλοποιήσουμε ό,τι περιγράψαμε!
- ❏ Μπορούμε να εισάγουμε κατά βούληση φωτεινές πηγές, αντικείμενα (σφαίρες..) , να τις χρωματίσουμε όπως θέλουμε, να αλλάξουμε τα γεωμετρικά χαρακτηριστικά τους, κλπ.

Ανακεφαλαίωση

- ❏ Είδαμε τις βασικές αρχές σύμφωνα με τις οποίες μοντελοποιούμε το φως σε έναν raytracer
- ❏ Τρία είδη πηγών φωτός - σημειακά, κατευθυνόμενα, περιβάλλον
- ❏ Είδαμε πως μεταφράζουμε τα παραπάνω, διακρίνοντας δύο περιπτώσεις: “Ματ” επιφάνειες (= διάχυτη ανάκλαση) και γυαλιστερές επιφάνειες (=κατοπτρική ανάκλαση)
- ❏ Εξίσωση που μας δίνει την φωτεινή ένταση ανά σημείο:

$$I_P = I_A + \sum_{i=1}^n I_i \cdot \left[\frac{\langle \vec{N}, \vec{L}_i \rangle}{|\vec{N}| |\vec{L}_i|} + \left(\frac{\langle \vec{R}_i, \vec{V} \rangle}{|\vec{R}_i| |\vec{V}|} \right)^s \right]$$

- ❏ Στο επόμενο: Σκιές!

Μελέτη στο βιβλίο - πηγές

- ❏ G.Gambetta, Basic Raytracing (<https://gabrielgambetta.com/computer-graphics-from-scratch/03-light.html>)
- ❏ (Αυτό το κεφάλαιο υπάρχει και στο eclass ως pdf).

- ❏ Υπάρχει πιο σύνθετος τρόπος να μοντελοποιήσουμε το φως και την ανάκλαση;;
 - ❏ Ρίξτε μια ματιά π.χ. στο Yang et al., «Light Sampling Field and BRDF Representation for Physically-based Neural Rendering», International Conference on Learning Representations, 2023 (<https://arxiv.org/abs/2304.05472>) για μερικές ιδέες ...

Ερωτήσεις ... ;;;

