

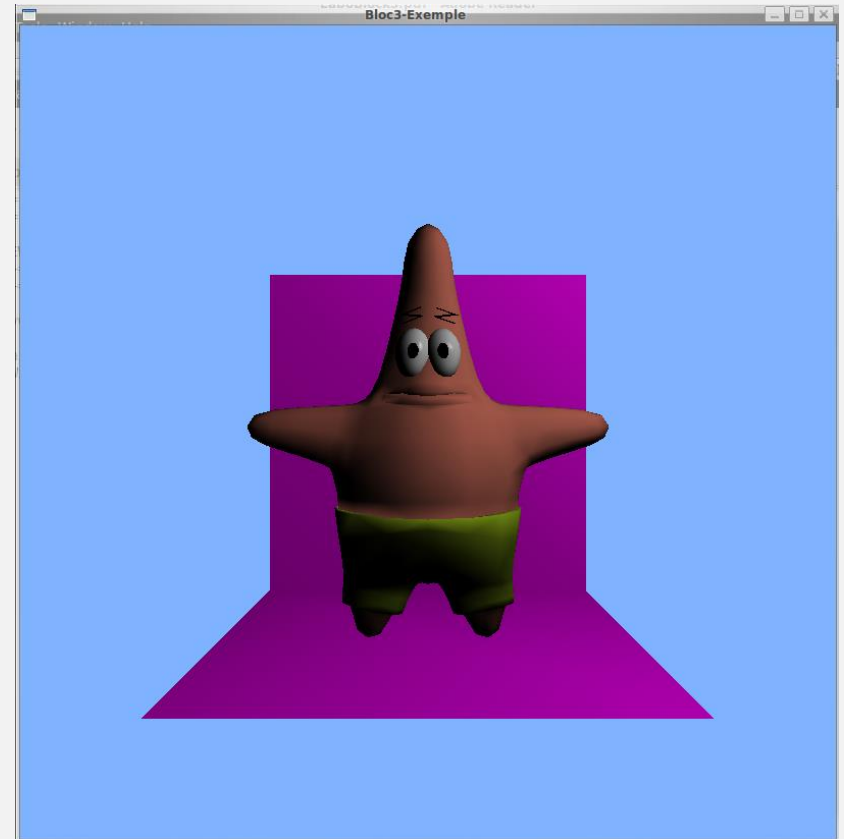
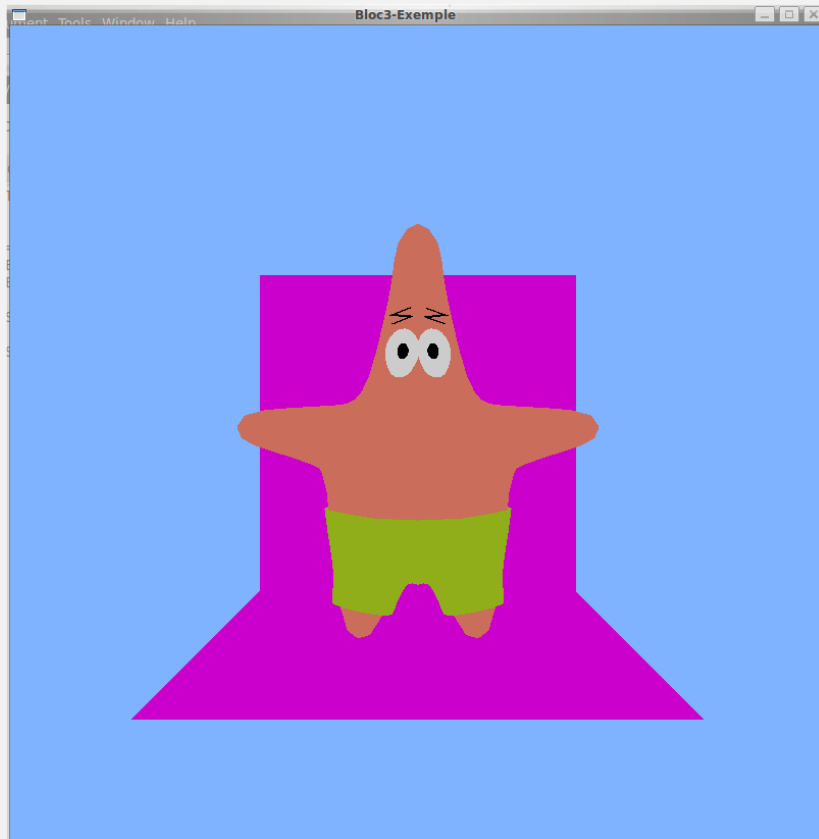
# Laboratori INDI

Bloc 3 OpenGL

Sessió 3.1

# Laboratori OpenGL – Sessió 3.1

## Realisme - Il·luminació:



# Taula de continguts

1. Recordatori il·luminació

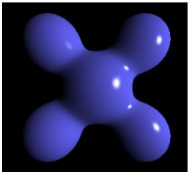
2. SC i informació dels models empírics

3. Primers passos

4. Exercicis

# Càlcul color en un punt: models empírics

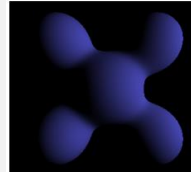
$$I_{\lambda}(P) = I_{a\lambda} k_{a\lambda} + \sum_i (I_{fi\lambda} k_{d\lambda} \cos(\Phi_i)) + \sum_i (I_{fi\lambda} k_{s\lambda} \cos^n(\alpha_i))$$



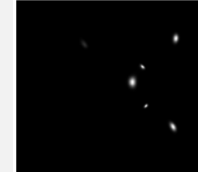
Phong Reflection



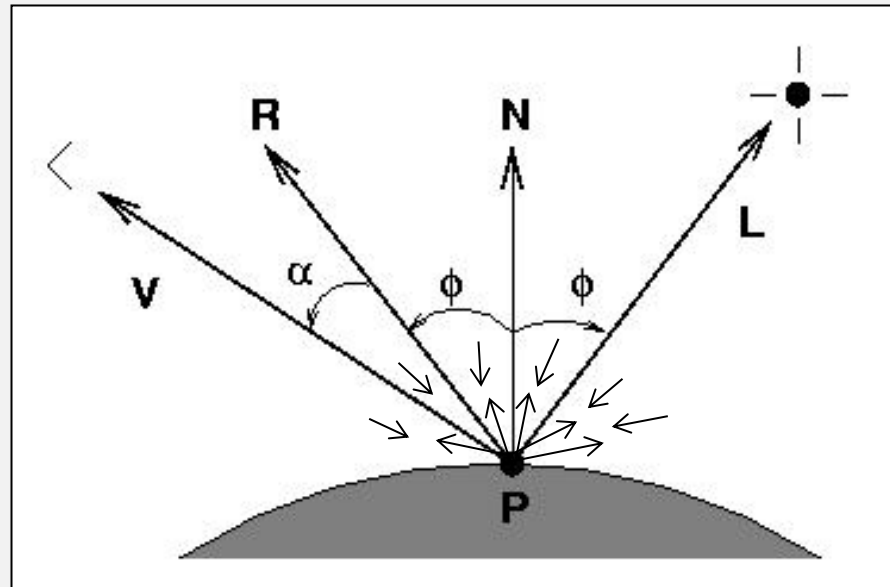
Ambient



Diffuse



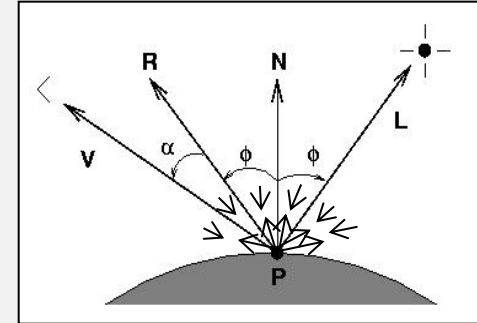
Specular



# Càlcul color en un punt: models empírics

## Què necessitem?

- Propietats del material
  - Vector normal
  - Color de llum ambient
  - Posició del focus de llum
  - Color del focus de llum
  - Posició observador – en SCO sabem que és (0,0,0) -
- Per cada vertex (punt)
- Per cada focus de llum



$$I_{\lambda}(P) = I_{a\lambda} k_{a\lambda} + \sum_i (I_{fi\lambda} k_{di\lambda} \cos(\Phi_i)) + \sum_i (I_{fi\lambda} k_{si\lambda} \cos^n(\alpha_i))$$

# Taula de continguts

1. Recordatori il·luminació

2. SC i informació dels models empírics

3. Primers passos

4. Exercicis

# Sistemes de coordenades

- Quins sistemes de coordenades coneixem?

– SC **M**odel

– SC **A**plicació

– SC **O**bservador

– SC **C**lipping

– SC **D**ispositiu

**TG\*** (Model matrix)

**VM\*** (View matrix)

**PM\*** (Project matrix)

**Triarem SCO doncs:**

- Els objectes “estan a lloc” ( a SCM no)
- L’observador està al (0,0,0) ( a SCA no)
- No hi ha distorsió angular a l’escena. (a SCC hi ha distorsió)

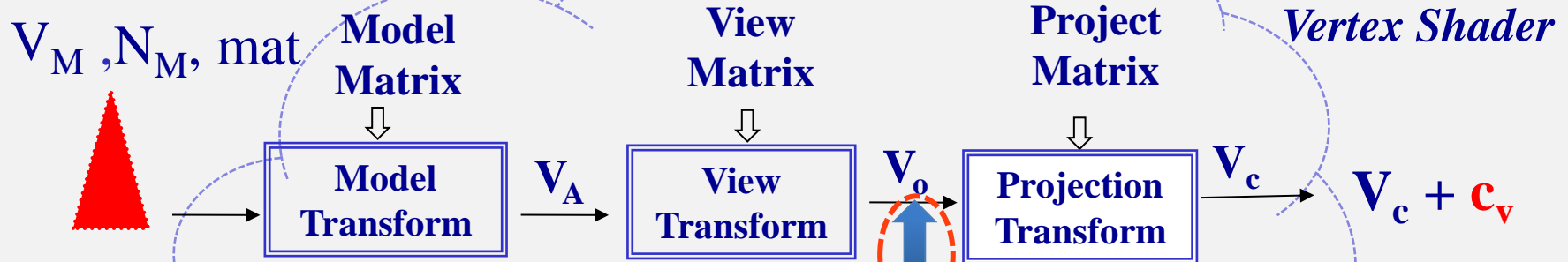
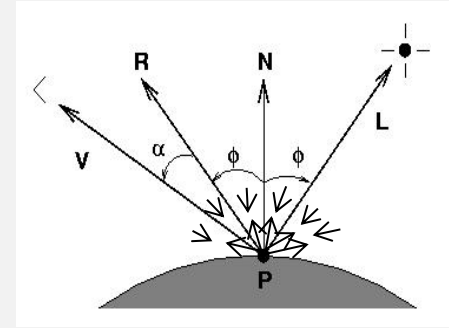
# Càlcul color en Vèrtex Shader

## - Atributs de vèrtex:

Coordenades vèrtex, normal i constants de material per vèrtex en VBOs dels VAO

## - *Uniforms o inicialitzats directament en VS:*

- Fonts de llum actives => color, posició
- Llum ambient



Càlcul del **color** en els vèrtexs

$$= I_{a\lambda} k_{a\lambda} + \sum_i (I_{fi\lambda} k_{d\lambda} \cos(\Phi_i)) + \sum_i (I_{fi\lambda} k_{s\lambda} \cos^n(\alpha_i))$$



# Càlcul de color al Vertex Shader

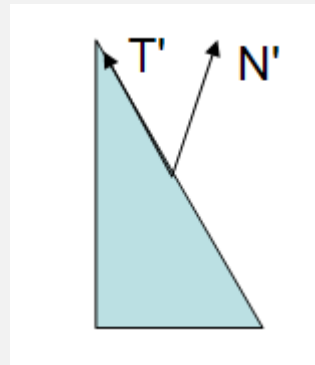
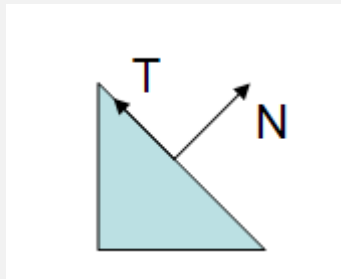
El càlcul el farem per cada vèrtex i **el farem en SCO**.

Per tant:

- Cal passar la posició del vèrtex a SCO:
  - multiplicant per (**view \* TG**)
- La posició del focus de llum també ha d'estar en SCO
  - Multiplicada per (**view**) (si no la tenim directament en SCO)
  - o Multiplicada per (**view\*TG**) (si la llum és mòbil)

# Càlcul color en un punt: models empírics

- Cal passar el vector normal a SCO
  - Si multipliqueu directament per **(view\*TG)**, no funcionarà si TG inclou escalats no uniformes, ja que alteren les normals:



**La normal deixa de ser perpendicular!**

- Ho solucionem multiplicant per la matriu **transposada** de la **inversa** de **(view \* TG)**.

-li direm **NormalMatrix-**

$$\text{NormalMatrix}(M) = (M^{-1})^T$$

# Càlcul color en un punt: models empírics

- Calcular matriu *transposada* de la *inversa* de ( $view * TG$ )

- Al vertex shader (en GLSL):

```
mat3 NormalMatrix = transpose(inverse(mat3(view*TG)));
```

- es fa el càlcul de la matriu per a cada vèrtex...és poc eficient

- Al programa (amb glm):

```
glm::mat3 NormalMatrix = glm::inverseTranspose(glm::mat3(View*TG));
```

- cal tenir les matrius View i TG com a atributs de la classe

- `#include "glm/gtc/matrix_inverse.hpp"`

- i cal passar la matriu com a *uniform* al VS per cada objecte

# Taula de continguts

1. Recordatori il·luminació
2. SC i informació dels models empírics
3. Primers passos
4. Exercicis

# Anàlisi del codi de l'esquelet

- Analitzar quins mètodes implementats.
- Analitzar implementació dels mètodes.
  - Quina càmera tenim?
  - Quina escena?
  - Quina interacció?
- Analitzar els shaders
  - Atributs, uniforms, mètodes

# Taula de continguts

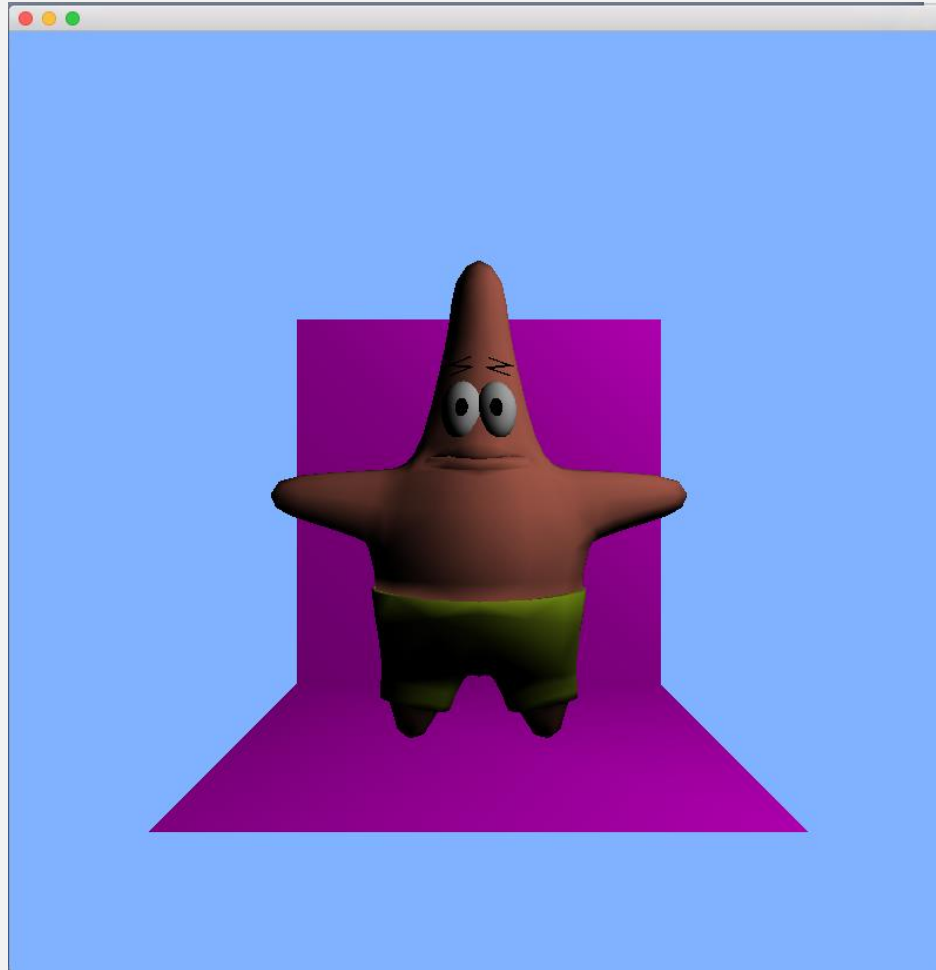
1. Recordatori il·luminació
2. SC i informació dels models empírics
3. Primers passos
4. Exercicis

# Exercici 1

## Càlcul color usant model Lambert:

```
vec3 Lambert (vec3 NormSCO, vec3 L)
{
    // Aquesta funció calcula la il·luminació amb Lambert assumint que els vectors
    // que rep com a paràmetres estan normalitzats
    vec3 colRes = llumAmbient * matamb; // Inicialitzem color a component ambient
    // Afegim component difusa, si n'hi ha
    if (dot (L, NormSCO) > 0)
        colRes = colRes + colFocus * matdiff * dot (L, NormSCO);
    return (colRes);
}
```

Cal calcular en *main*: L en SCO, Normal en SCO,  
normalitzar vectors i cridar a Lambert



Proveu a moure càmera (si voleu poseu tb rotació en X) i veureu que cares il·luminades no varien. **Llum d'escena.**

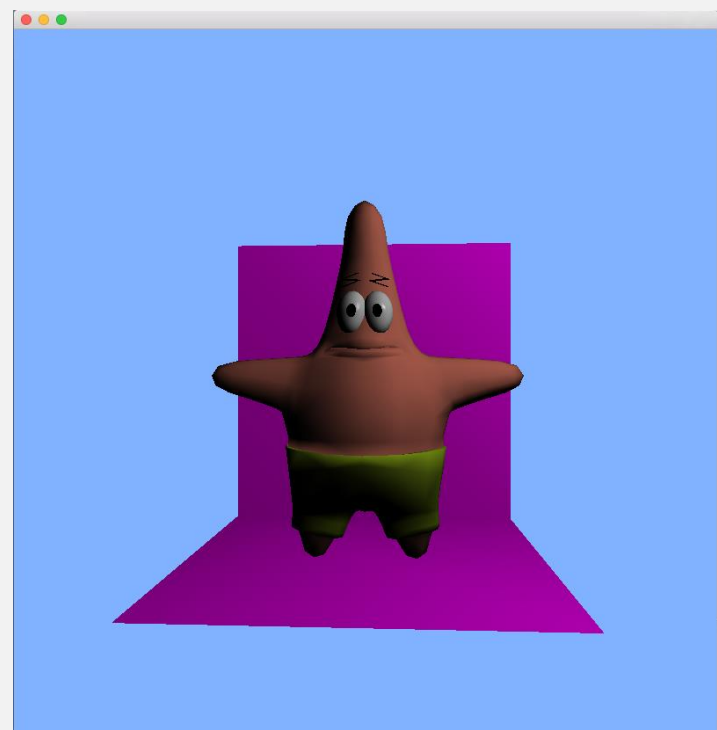
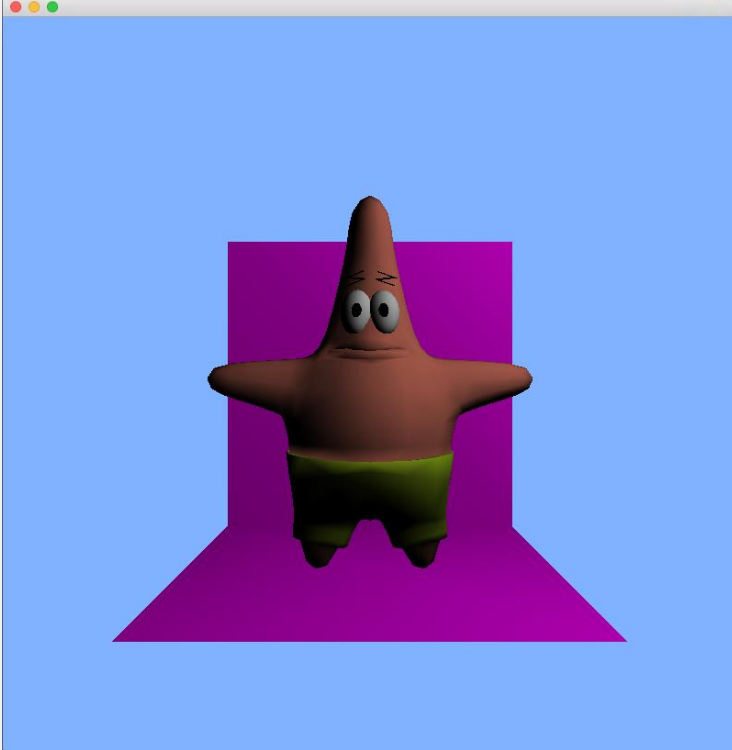


# Exercici 2

## Càlcul color usant model Phong:

**vec3 Phong (vec3 NormSCO, vec3 L, vec4 vertSCO)**

```
{  
    // Els vectors rebuts com a paràmetres estan normalitzats  
    vec3 colRes = Lambert (NormSCO, L); // Inicialitzem color a Lambert  
    // Calculem R i V  
    if (dot (NormSCO, L) < 0)  
        return colRes; // no afecta la component especular  
    vec3 R = reflect (-L, NormSCO); // equival a:: normalize (2.0 * dot (NormSCO, L) * NormSCO - L);  
    vec3 V = normalize (-vertSCO.xyz);  
    if ((dot (R, V) < 0) || (matshin == 0))  
        return colRes; // no afecta la component especular  
    // Afegim la component especular  
    float shine = pow (dot (R, V), matshin);  
    return colRes + matspec * colFocus * shine;  
}
```



Proveu a moure càmera (si voleu poseu tb rotació en X) i veureu que cares il·luminades no varien; però taca especular sí (en ulls).

**Llum d'escena.**

El terra pot tenir taca especular?

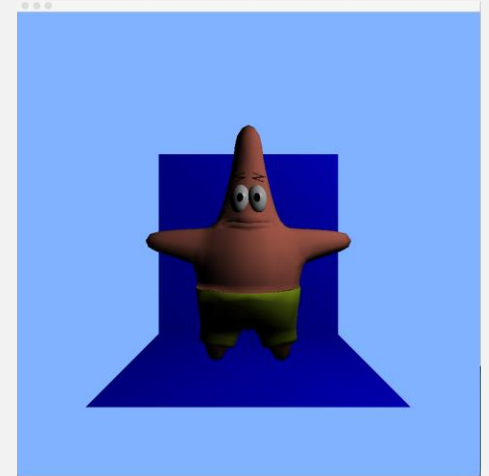
# Exercicis 3 i 4

3) Canvi material terra+paret

- Ha de ser de plàstic blau brillant

4) Canvi posició focus de llum

- Ha de ser la posició (1, 0, 1) en SCA



# Exercici 5

- Pas a uniforms de la posició i el color del focus de llum:

- Convertir la posició i el color en uniforms en el VS
- Inicialitzar aquests uniforms al MyGLWidget
- Fixem-nos que ara podríem passar el uniform de posició de la llum directament ja en SCO → estalvi en càlcul doncs el valor és igual per tots els vèrtex de l'escena.

- Podem també passar a uniform el color de la llum ambient

# Exercici 6

Fer que la posició del focus de llum es mogui amb les tecles K i L:

- K  $\rightarrow$  mou el focus cap a les X-
- L  $\rightarrow$  mou el focus cap a les X+

# Flat shading

- Proveu dir a rasterització que no interpoli color dels vèrtexs:
  - flat out fcolor

