

# Ray Tracing

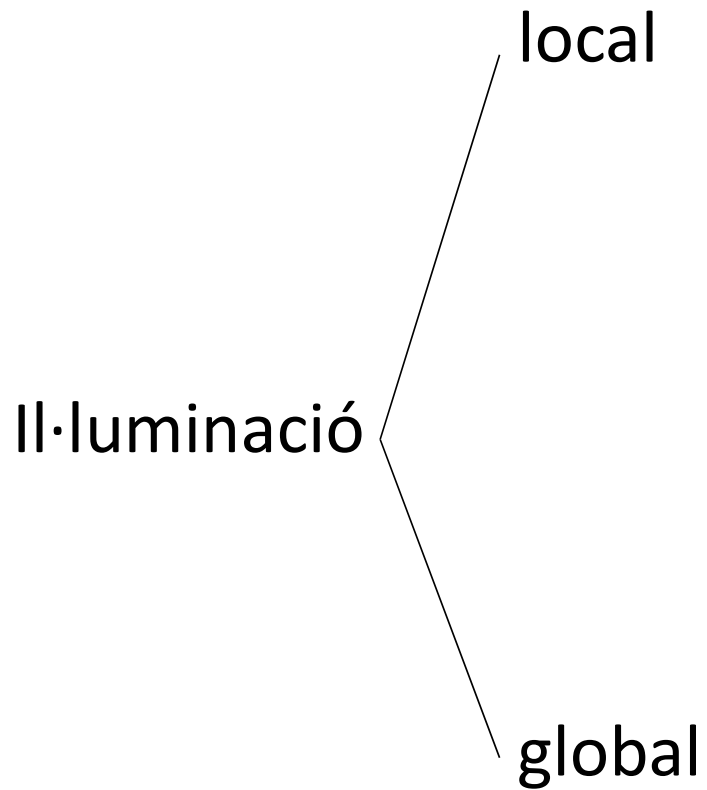
Carlos Andujar

Des 2022



# Il·luminació global

# Il·luminació local / global





Pabelló de Mies van der Rohe (foto: Ashley Pomeroy)

# Llum directa



Pabelló de Mies van der Rohe (model: eMirage, Claudio Andres, Hamza Cheggour)

# Llum directa + indirecta



Pabelló de Mies van der Rohe (model: eMirage, Claudio Andres, Hamza Cheggour)



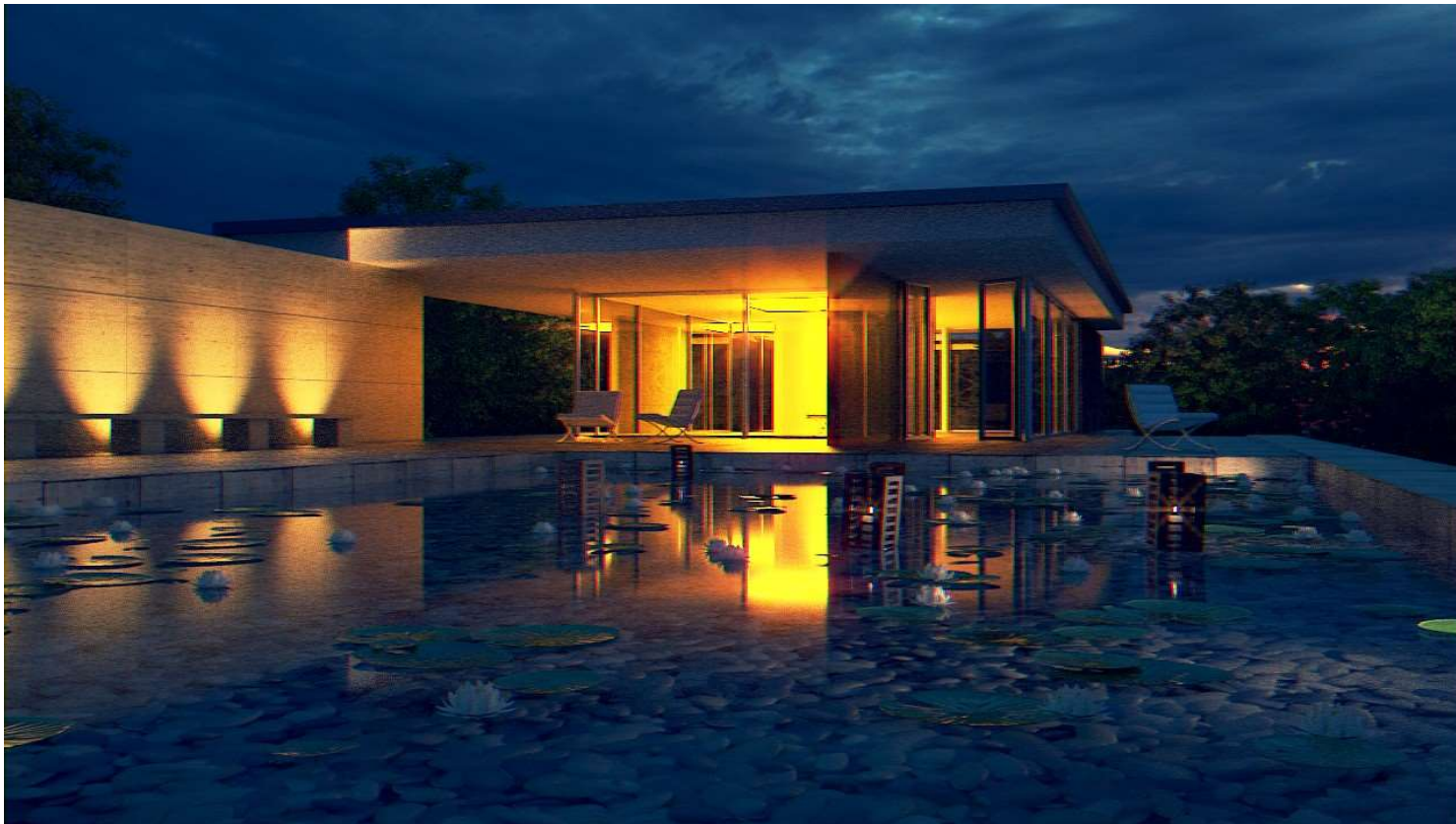
# Llum directa



Pabelló de Mies van der Rohe (model: eMirage, Claudio Andres, Hamza Cheggour)

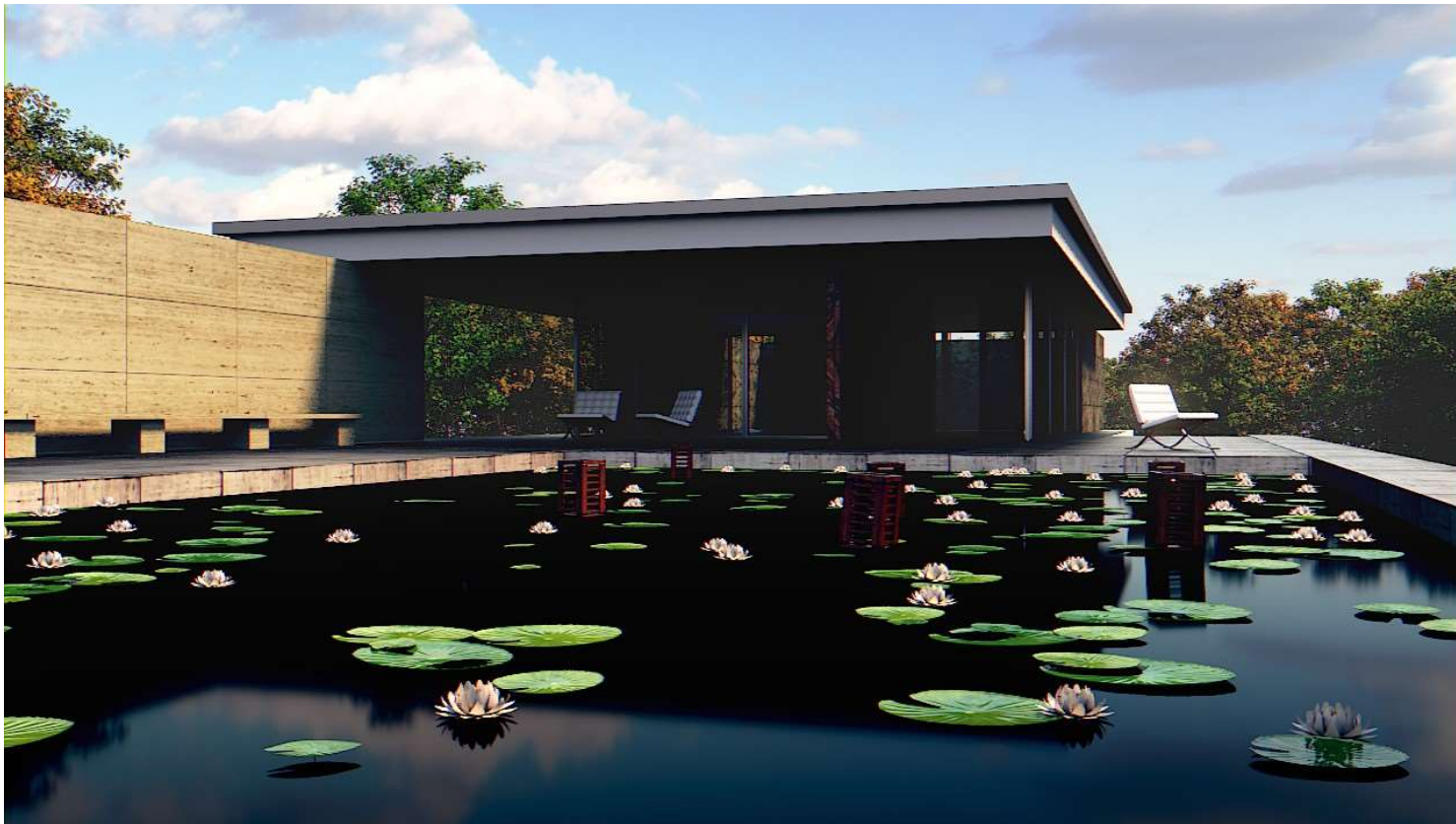


# Llum directa + indirecta



Pabelló de Mies van der Rohe (model: eMirage, Claudio Andres, Hamza Cheggour)

# Llum directa



Pabelló de Mies van der Rohe (model: eMirage, Claudio Andres, Hamza Cheggour)

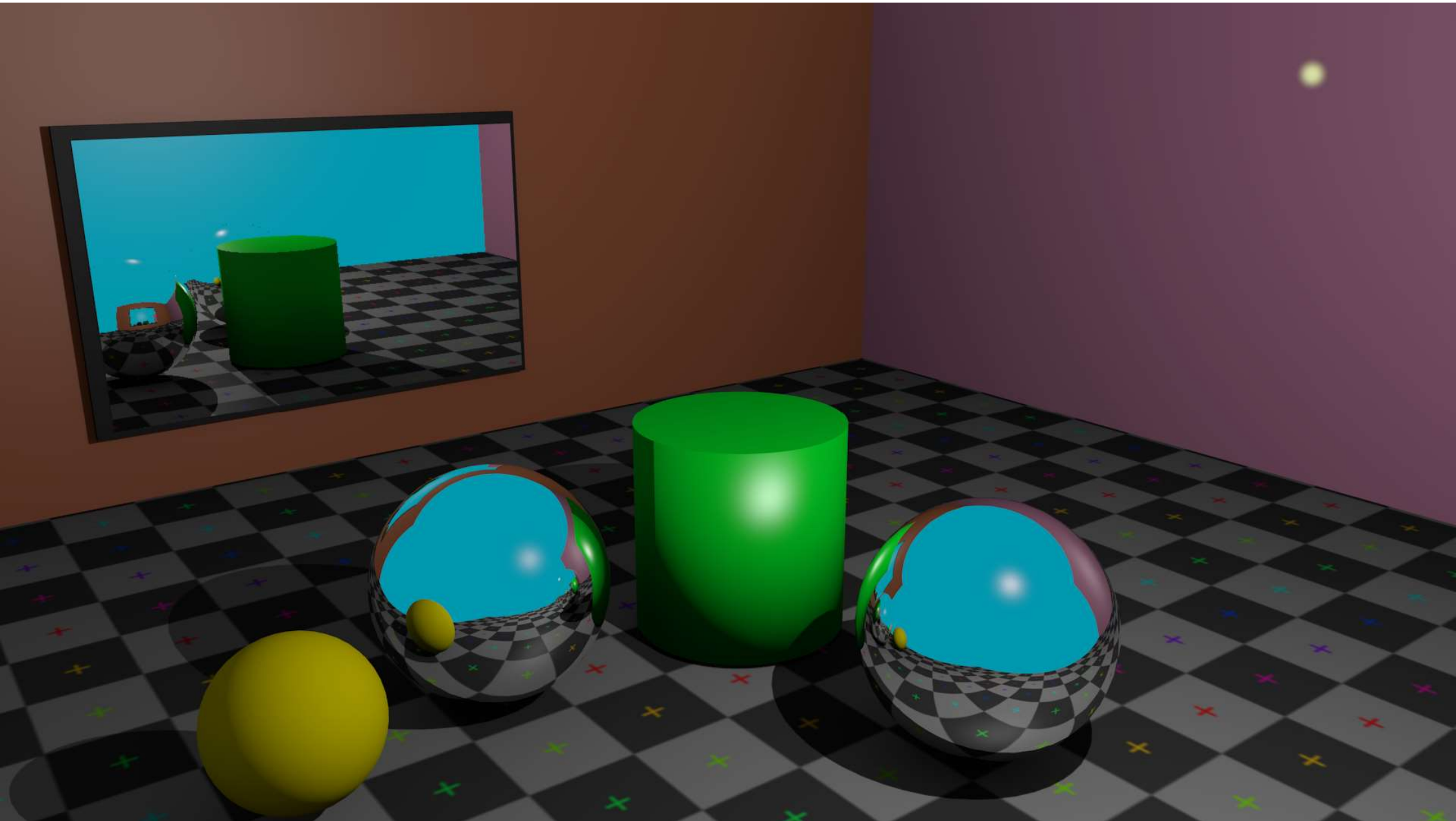
# Llum directa + indirecta

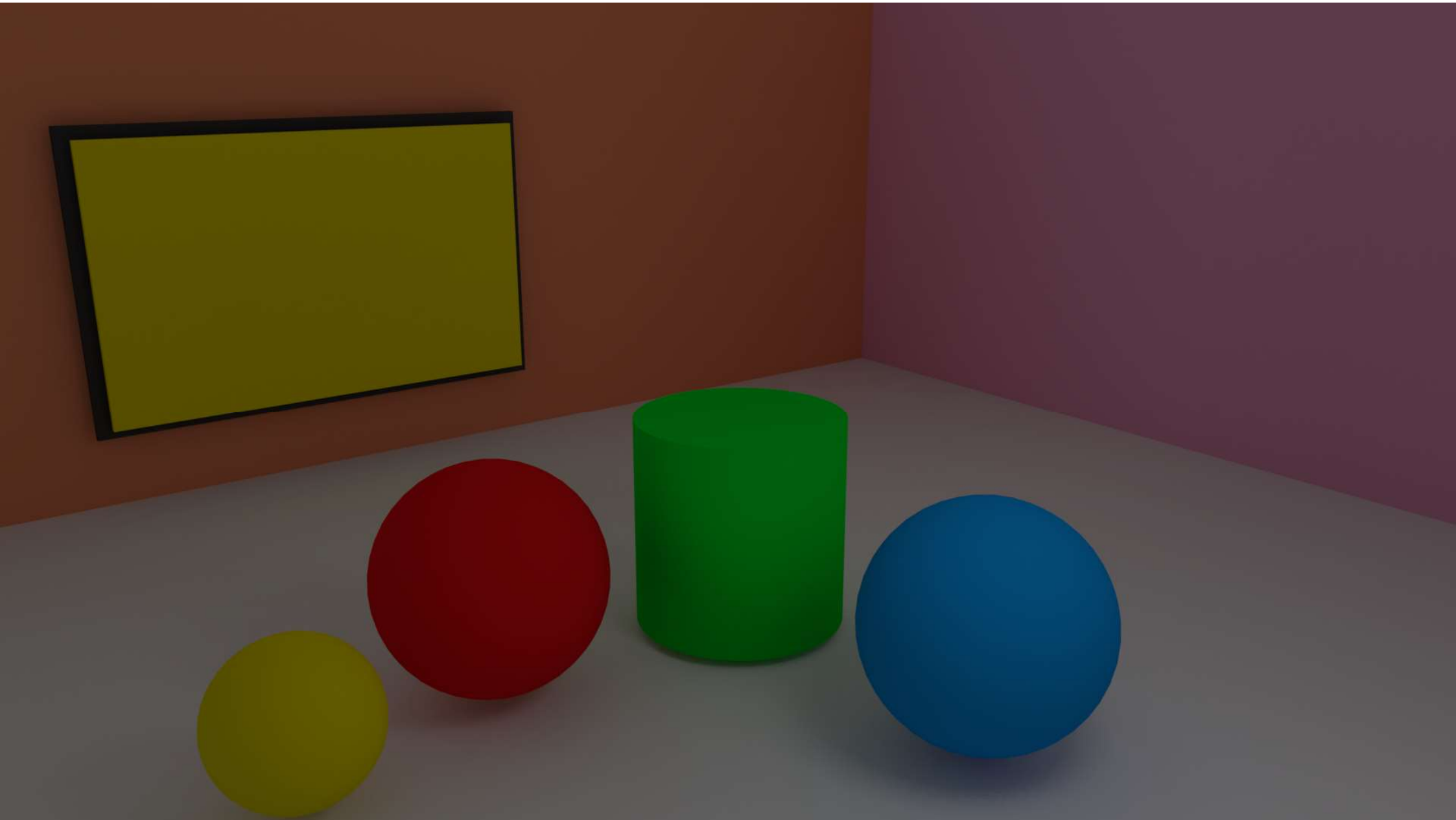


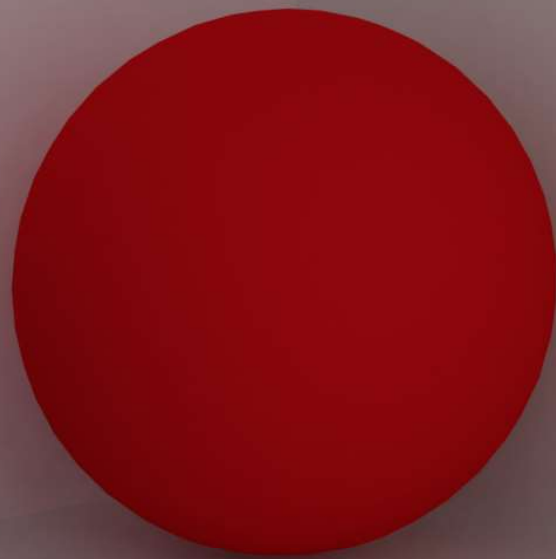
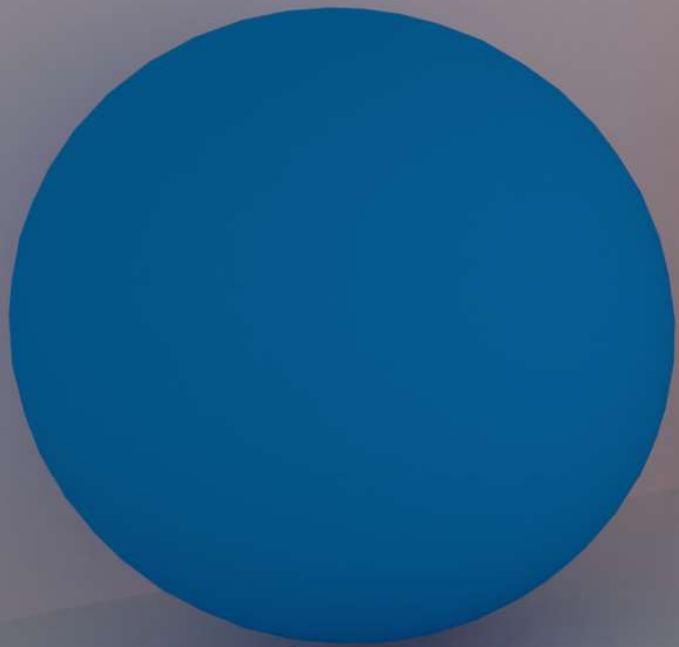
Pabelló de Mies van der Rohe (model: eMirage, Claudio Andres, Hamza Cheggour)

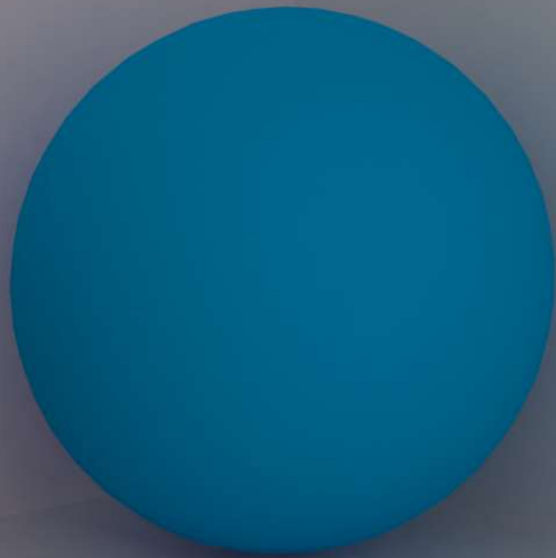
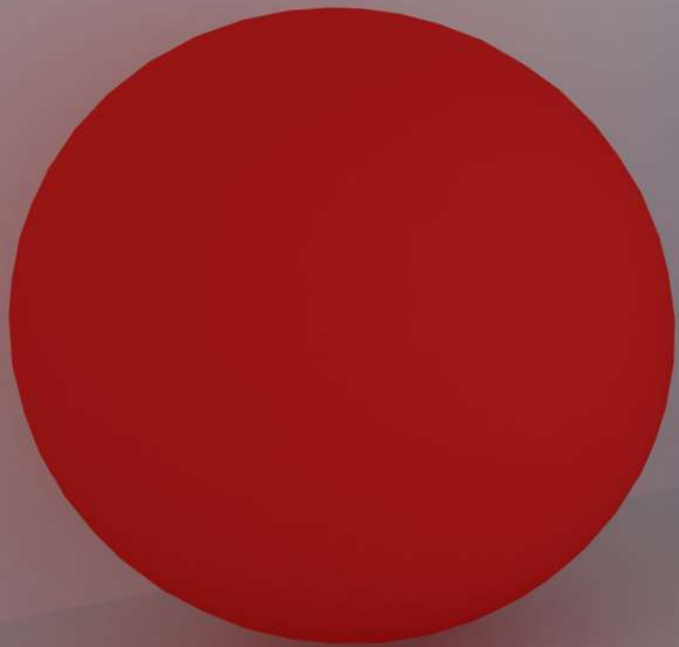


Sponza Atrium by Crytek













# RADIOMETRIA

# Mesurar el color de la llum

```
vec4 lightDiffuse = vec4(0.8, 0.8, 0.8, 1.0);  
vec4 lightSpecular = vec4(1.0, 1.0, 1.0, 1.0);
```

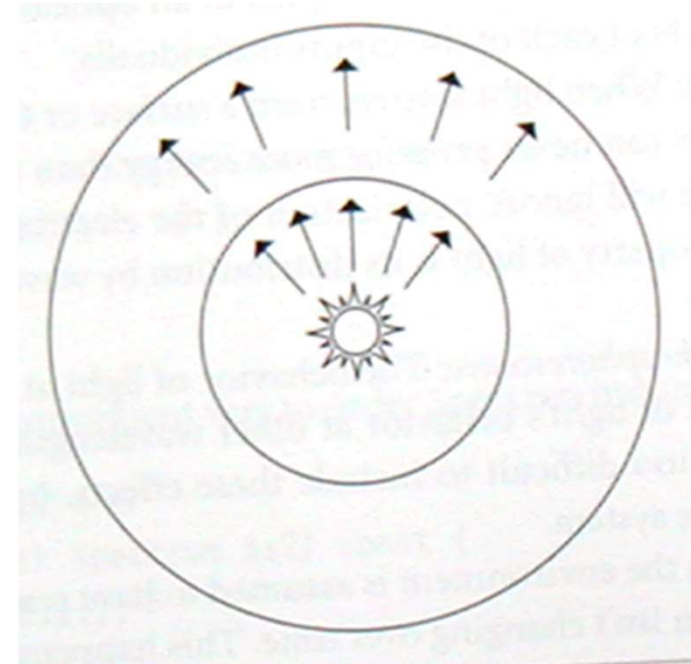


# Resum fotometria

Sím.	Radiomet.	Fotometria	Definició	Ús
$\Phi$	Fluxe (W)	Fluxe (lm)	Energia que travessa una superfície per unitat de temps	Energia total que emet una font de llum
E	Irradiancia (W/m <sup>2</sup> )	Iluminància (lux=lm/m <sup>2</sup> )	Fluxe per unitat <b>d'àrea</b>	Llum que incideix en un punt, des de qualsevol direcció
I	Intensitat (W/sr)	Intensitat (cd=lm/sr)	Fluxe per unitat <b>d'angle sòlid</b>	Distribució direccional d'una llum puntual
L	Radiància W/(sr·m <sup>2</sup> )	Luminància (cd/m <sup>2</sup> )	Fluxe per unitat <b>d'àrea</b> i unitat <b>d'angle sòlid</b>	Energia que travessa un punt en una determinada direcció

# Flux radiant $\Phi$

- Definició:
- Unitats:
- Exemple:
  
- Ús:



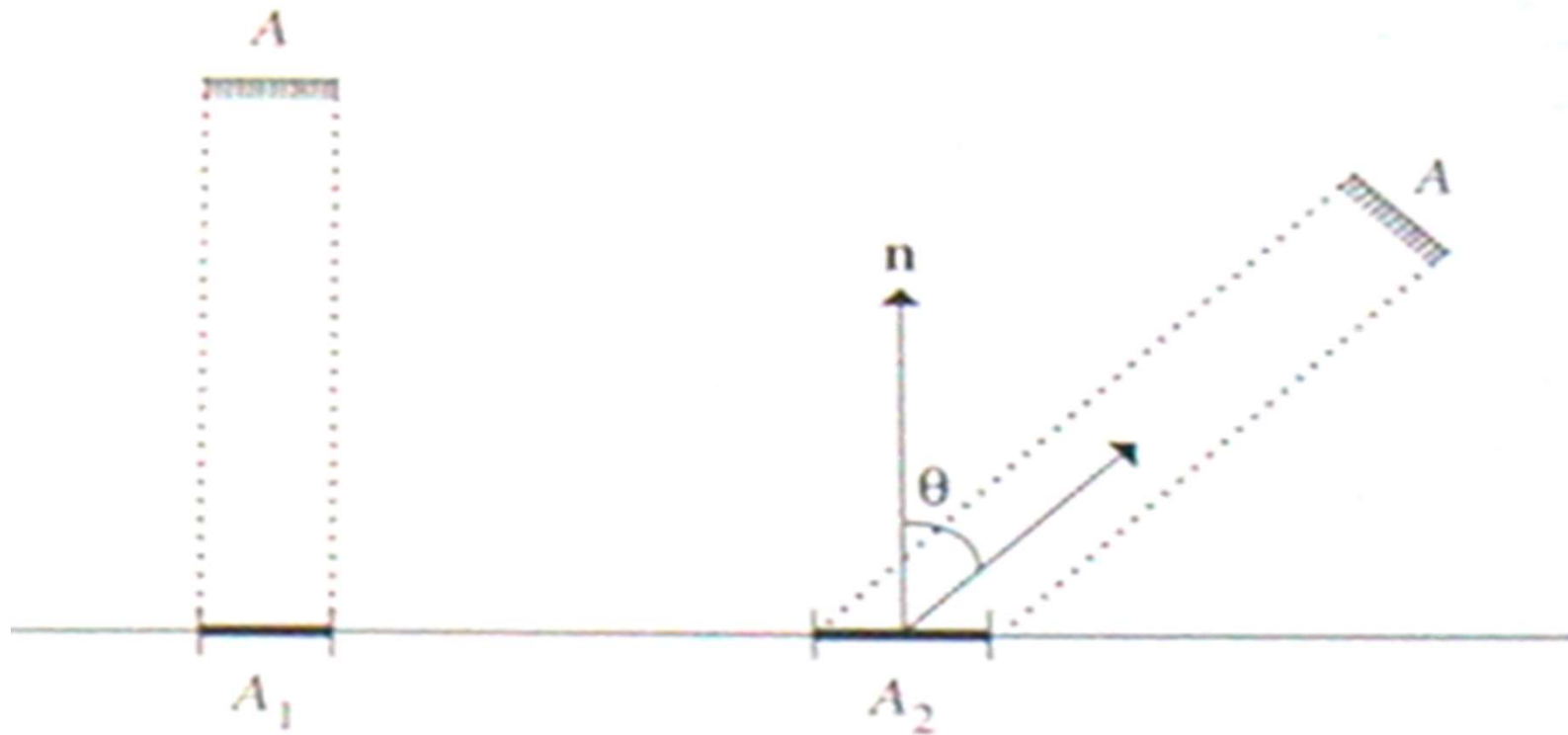
# Irradiància E

- Definició:
- Unitats:
- Exemple:
  
- Ús:

# Irradiància E



# Irradiància E i Lley de Lambert



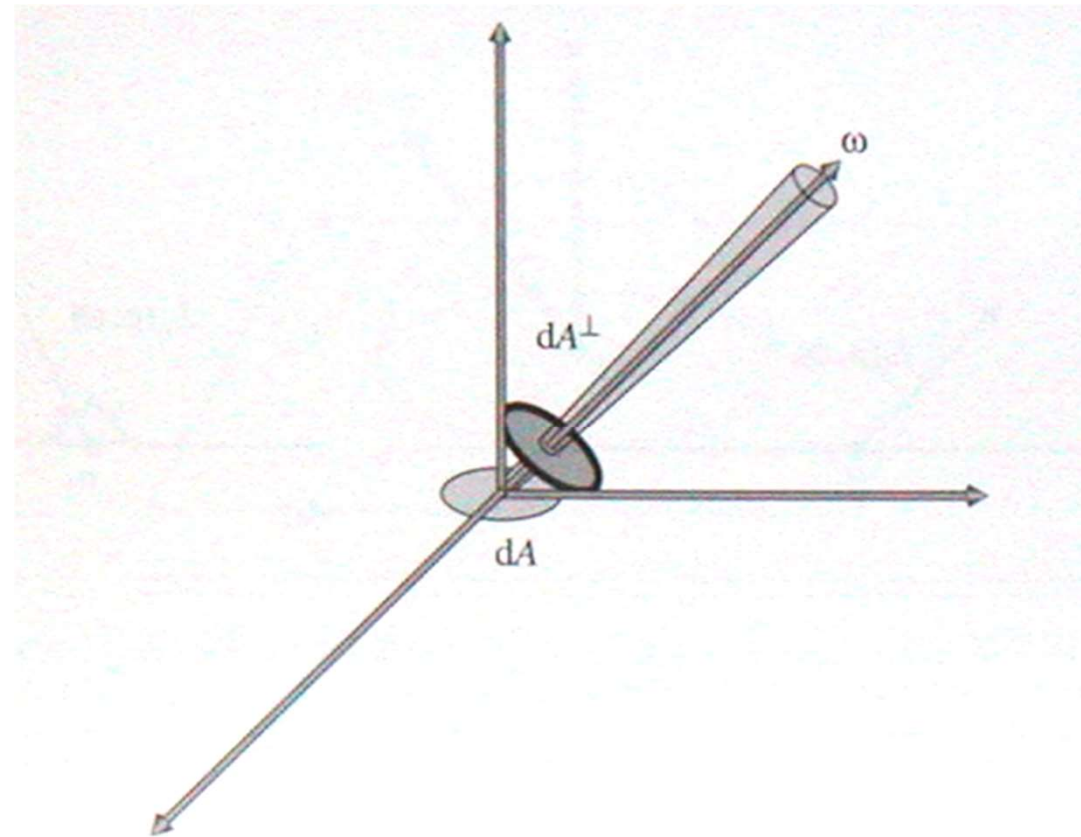
# Intensitat I

- Definició:
- Unitats:
- Exemple:
  
- Ús:



# Radiància $L(p, \omega)$

- Definició:
- Unitats:
- Exemple:
  
- Ús:



# Radiància $L(p,\omega)$

Propietats:

**BRDF – BTDF – BSDF**

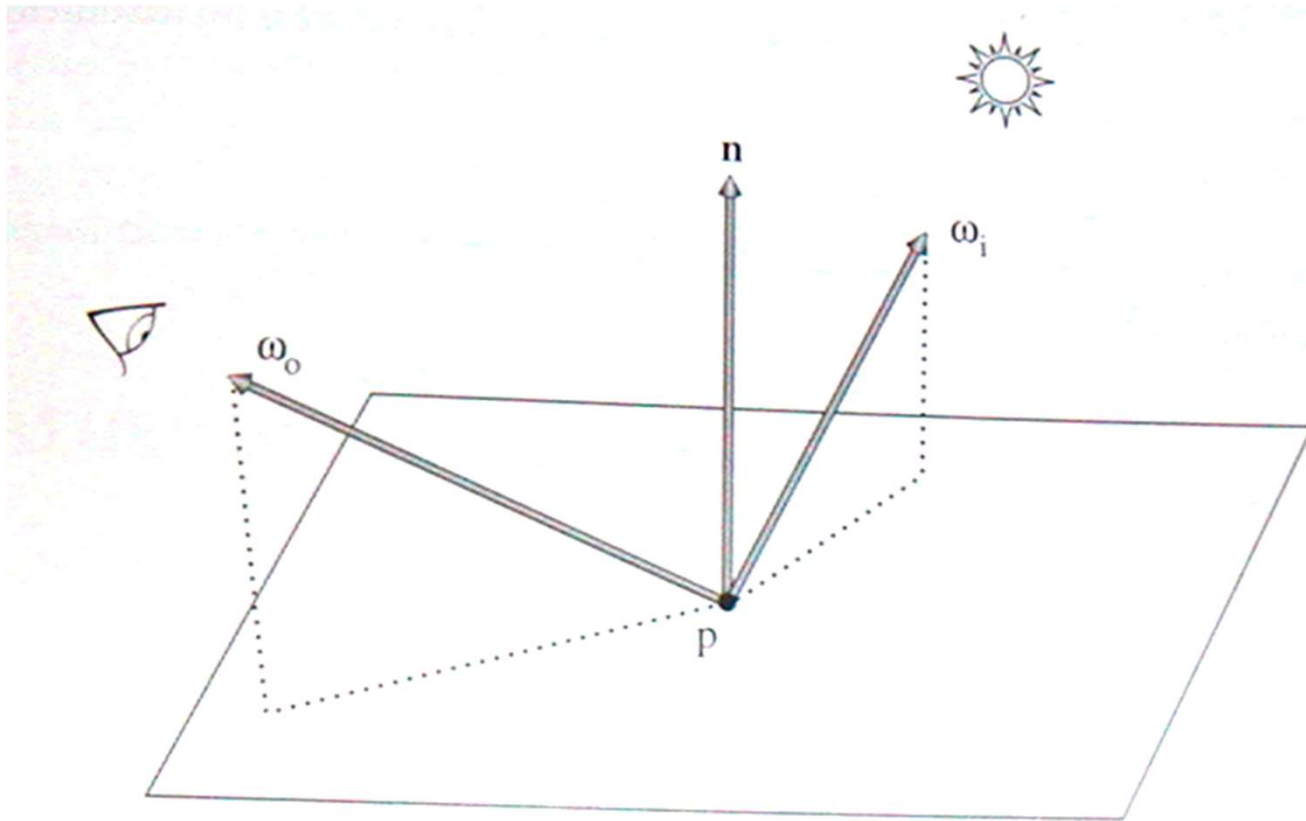
The image features a close-up, dense arrangement of Christmas ornaments. The ornaments are primarily in shades of gold, bronze, and copper, with some red ones visible on the right side. They have a glossy, reflective surface that catches the light, creating bright highlights and soft shadows. The text 'BRDF – BTDF – BSDF' is superimposed in the upper center of the image in a bold, white, sans-serif font. The background is a soft, out-of-focus white, which makes the colorful ornaments stand out.

# Reflectivitat / transmissió d'una superfície

```
uniform vec4 lightAmbient; // similar a gl_LightSource[0].ambient
uniform vec4 lightDiffuse; // similar a gl_LightSource[0].diffuse
uniform vec4 lightSpecular; // similar a gl_LightSource[0].specular
uniform vec4 lightPosition; // similar a gl_LightSource[0].position
// (sempre estarà en eye space)

uniform vec4 matAmbient; // similar a gl_FrontMaterial.ambient
uniform vec4 matDiffuse; // similar a gl_FrontMaterial.diffuse
uniform vec4 matSpecular; // similar a gl_FrontMaterial.specular
uniform float matShininess; // similar a gl_FrontMaterial.shininess
```

BRDF  $f(p, \omega_o, \omega_i)$



$$\text{BRDF } f(p, \omega_o, \omega_i)$$

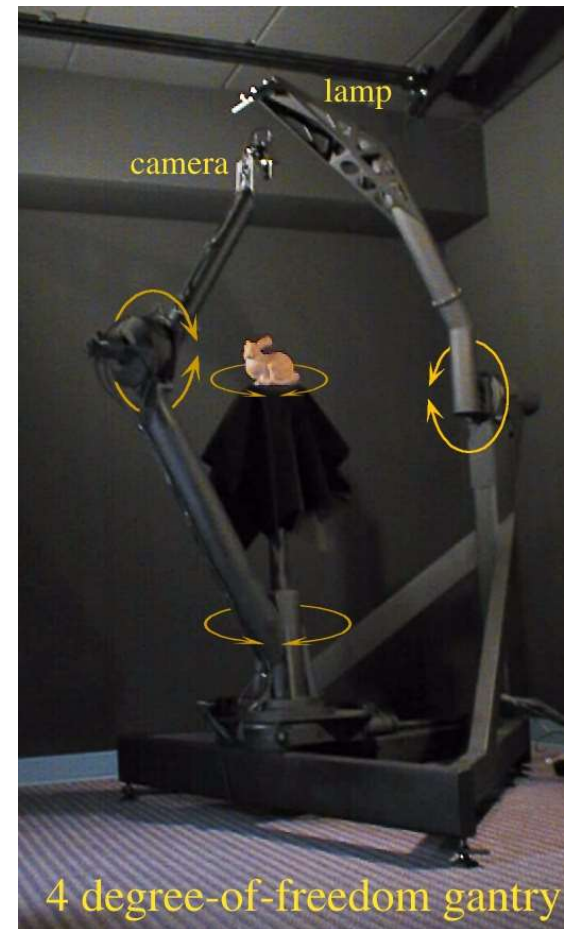
Propietats BRDFs basats en física:

# BRDF



# MERL BRDF database

- [www.merl.com/brdf/](http://www.merl.com/brdf/)





# BRDF de pinturas



# Phong vs BRDF mesurat



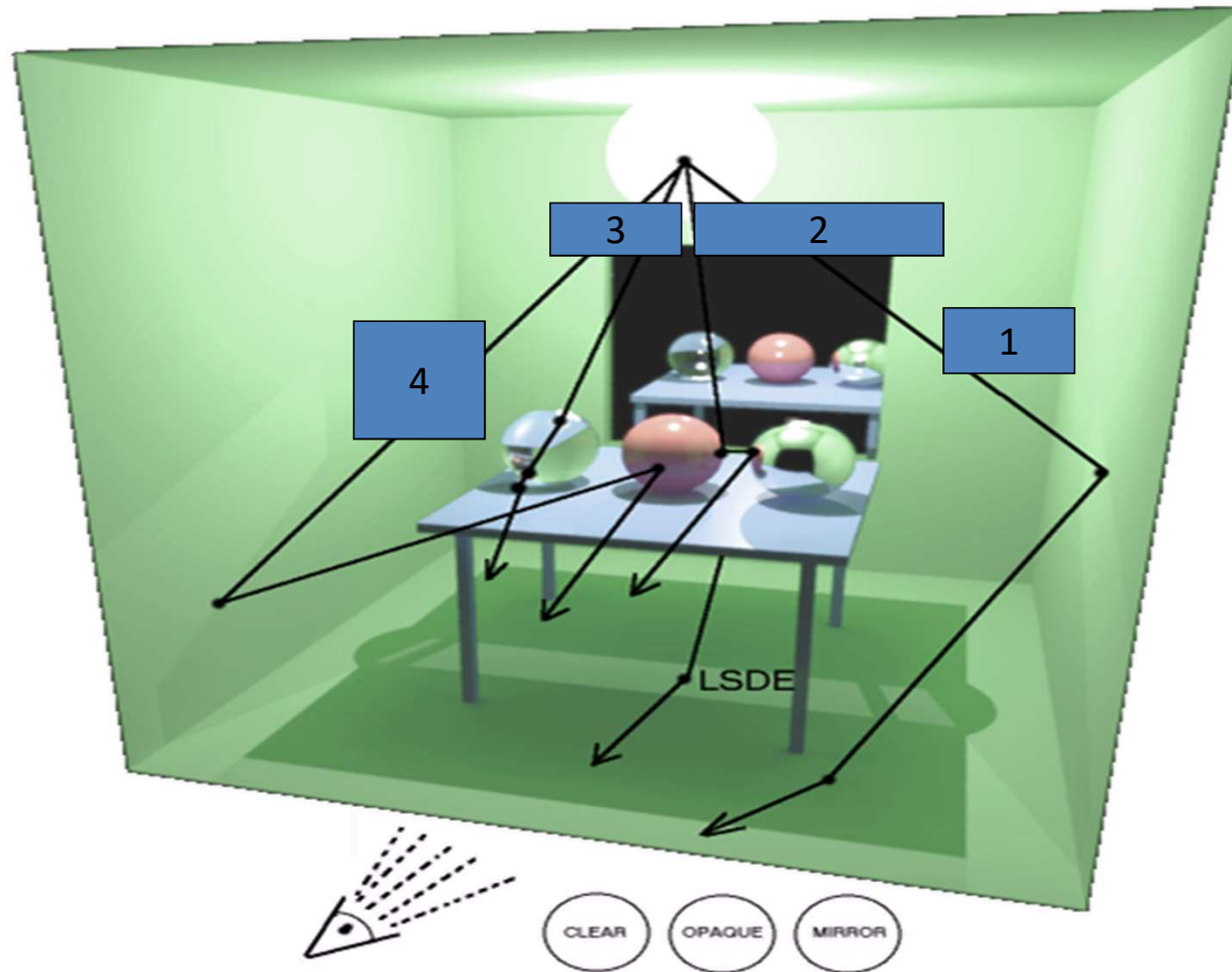


# EQÜACIÓ DEL RENDERING

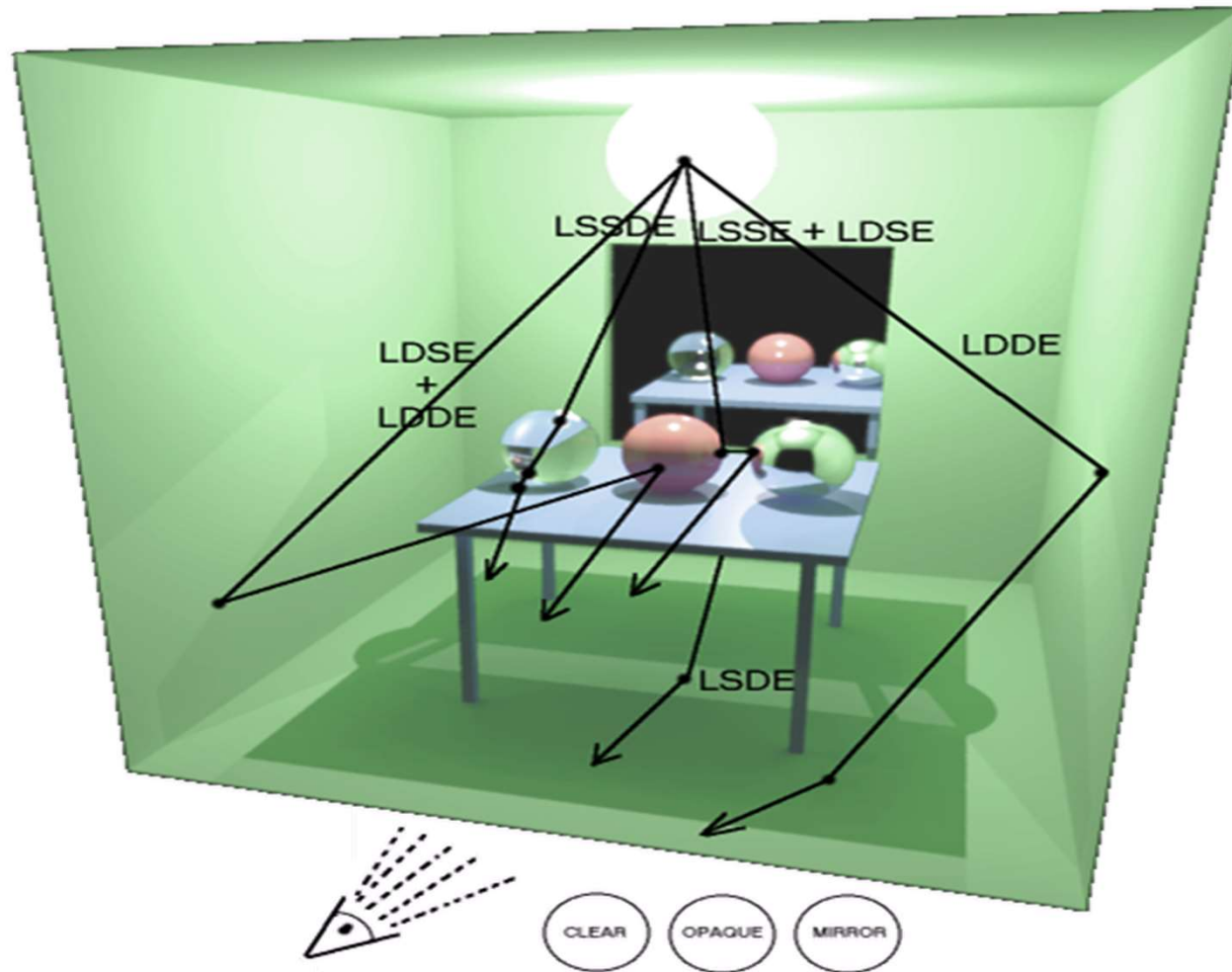
Equació general del rendering (Kajiya 1986)

# LIGHT PATHS

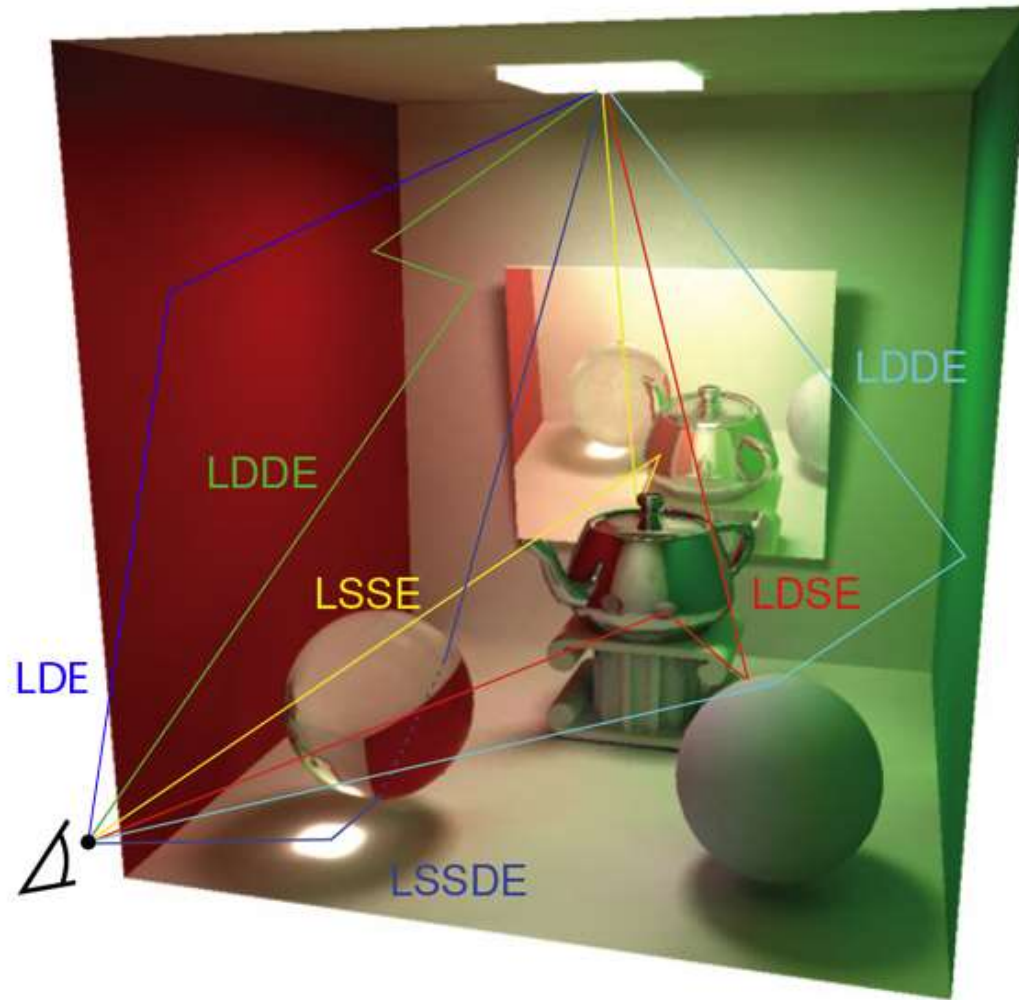
# Light paths



# Light paths

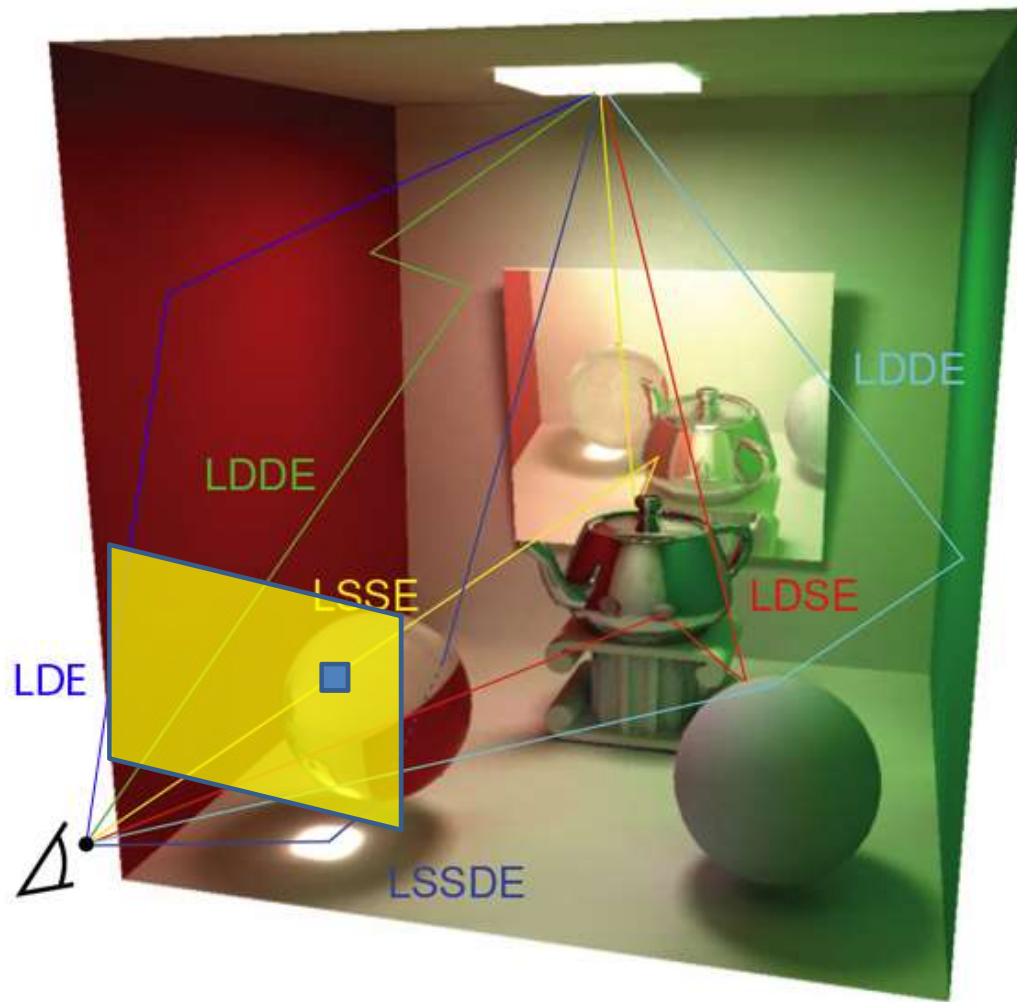


# Light paths





# Light paths



# **RAY CASTING - RAY TRACING**

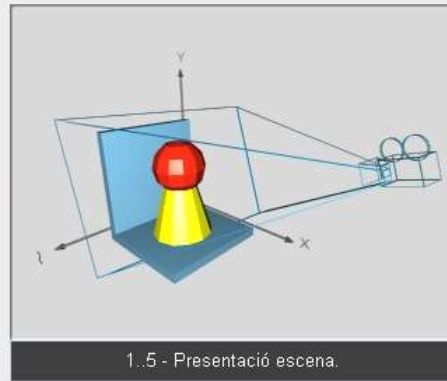
# Ray-casting

```
accio ray_casting
```

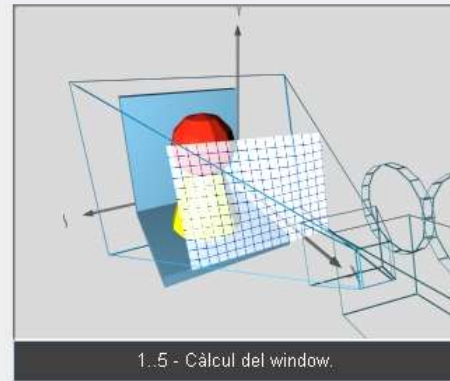
```
per y en [vyM...vym] fer  
per x en [vxM...vxM] fer  
  r:=calcul_raig(x,y,Obs)  
per cada cara fer  
  trobar_interseccions_raig_cara  
fper  
  ordenar_interseccions  
fper  
fper  
faccio
```

Anterior

Següent



1..5 - Presentació escena.



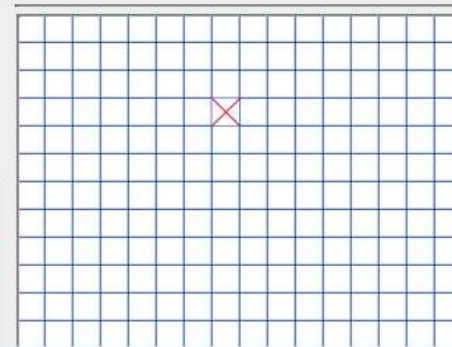
1..5 - Càlcul del window.

# Ray-casting

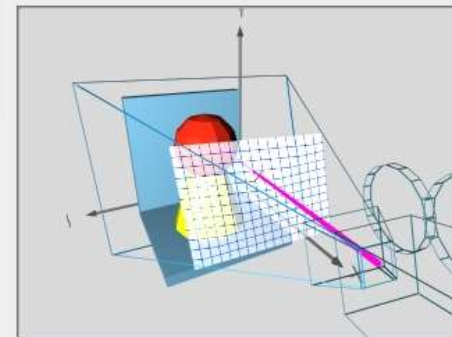
```
accio ray_casting
  per y en [vyM...vym] fer
    per x en [vxm...vxM] fer
      r:=calcul_raig(x,y,Obs)
      trobar_interseccions_raig_cara
    fper
  ordenar_interseccions
fper
fper
faccio
```

Anterior

Següent



2.5 - Donat un píxel.



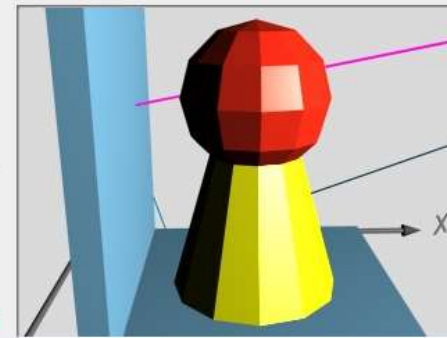
2.5 - Traçar raig per píxel.

# Ray-casting

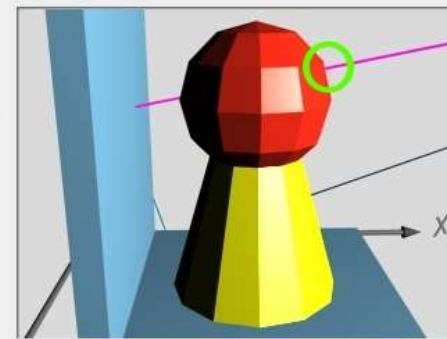
```
accio ray_casting
  per y en [vyM...vym] fer
    per x en [vxm...vxM] fer
      r:=calcul_raig(x,y,Obs)
      per cada cara fer
        trobar_interseccions_raig_cara
      fper
        ordenar_interseccions
      fper
        fper
        faccio
```

Anterior

Següent



3.5 - Càlcul interseccions.



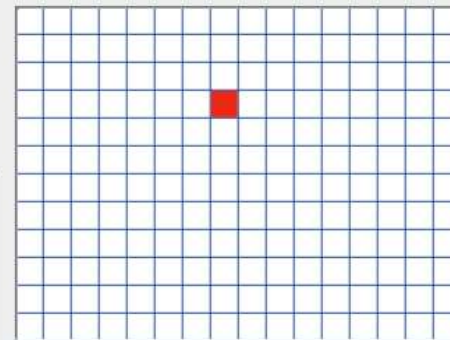
3.5 - Obtenir intersecció més propera.

# Ray-casting

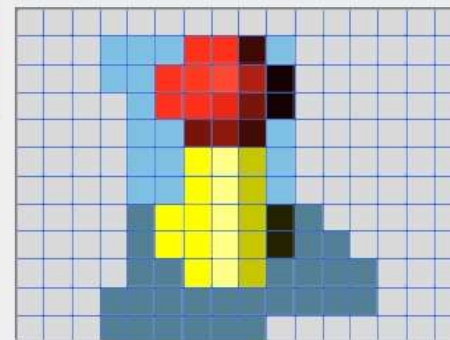
```
accio ray_casting
per y en [vyM...vym] fer
  per x en [vxm...vxM] fer
    r:=calcul_raig(x,y,Obs)
  per cada cara fer
    trobar_interseccions_raig_cara
  fper
    ordenar_interseccions
fper
fper
faccio
```

Anterior

Següent



4.5 - Pintar píxel del color de la intersecció.



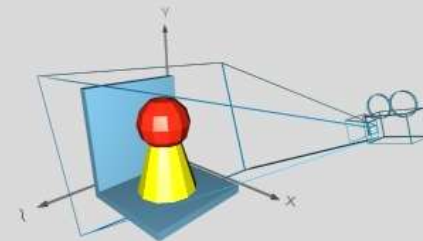
4.5 - Resultat en baixa resolució.

# Ray-casting

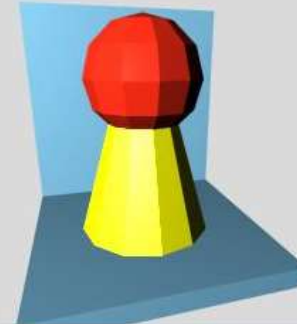
```
accio ray_casting
per y en [vyM...vym] fer
  per x en [vxm...vxM] fer
    r:=calcul_raig(x,y,Obs)
    per cada cara fer
      trobar_interseccions_raig_cara
    fper
      ordenar_interseccions
  fper
fper
  faccio
```

Anterior

Següent



5.5 - Presentació escena.



5.5 - Visió final en resolució estàndard.

# RAY TRACING



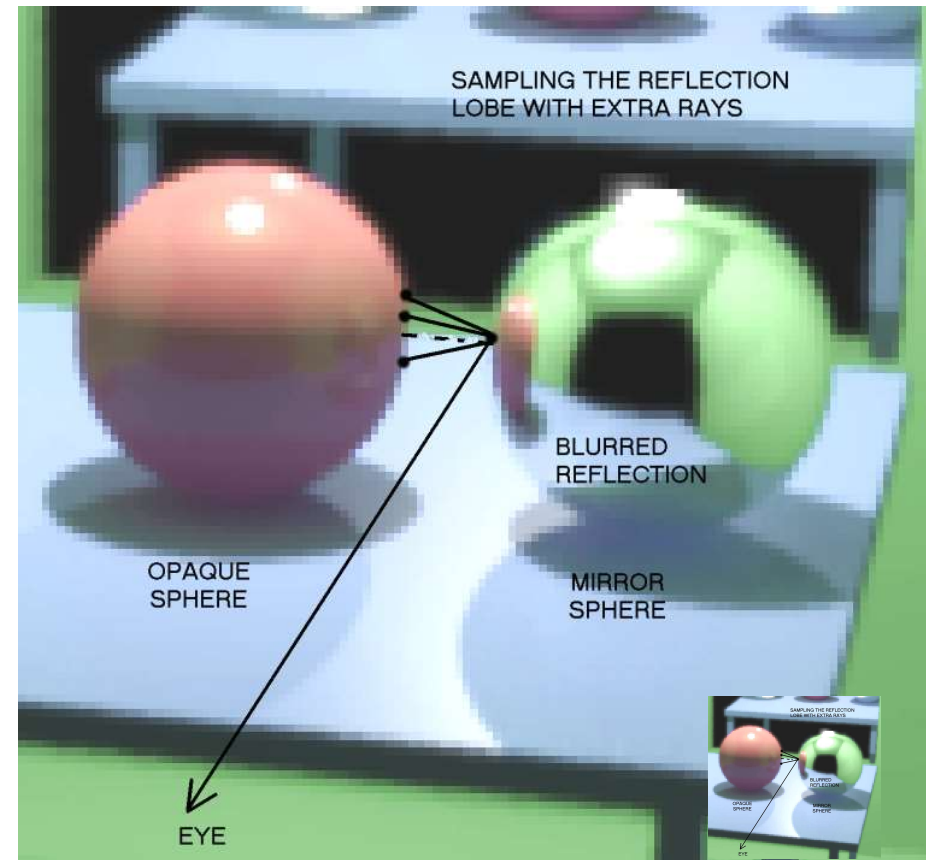
# Ray-tracing clàssic



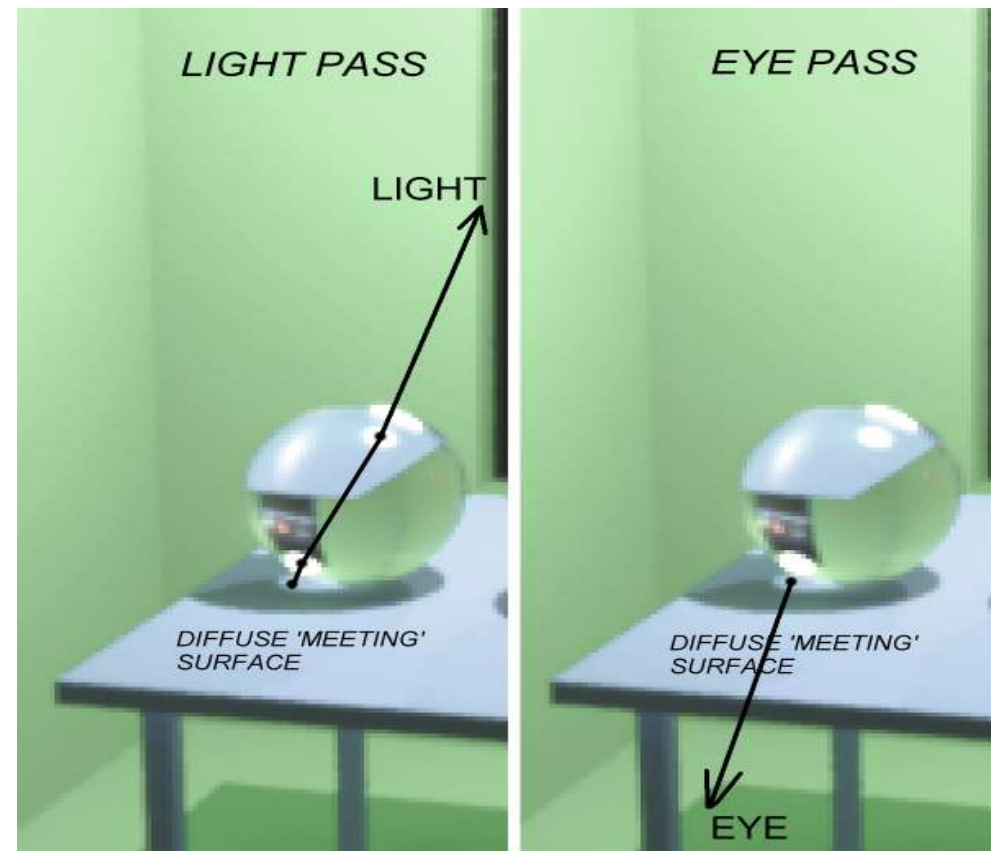
# Path tracing



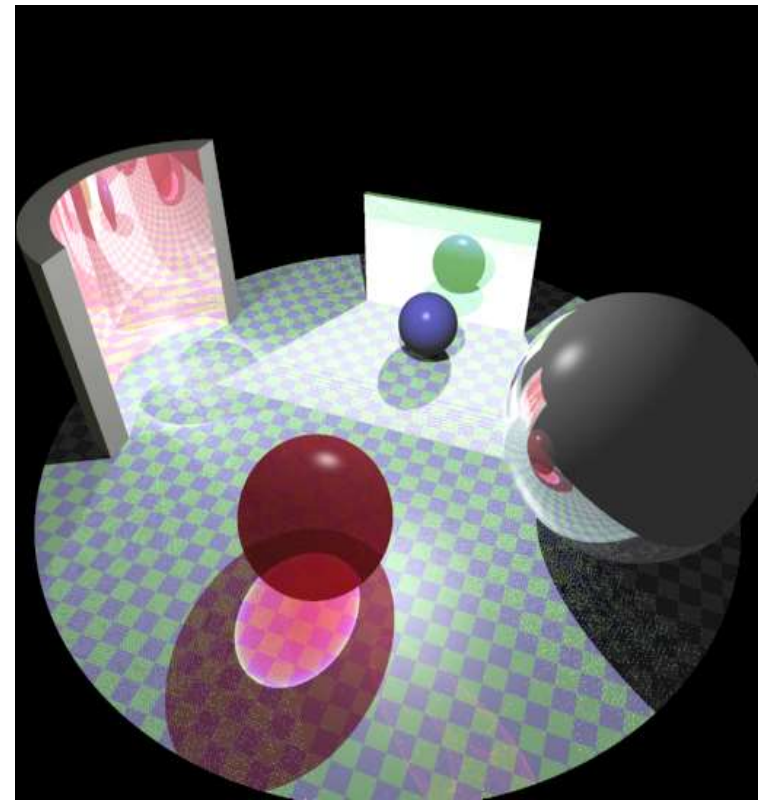
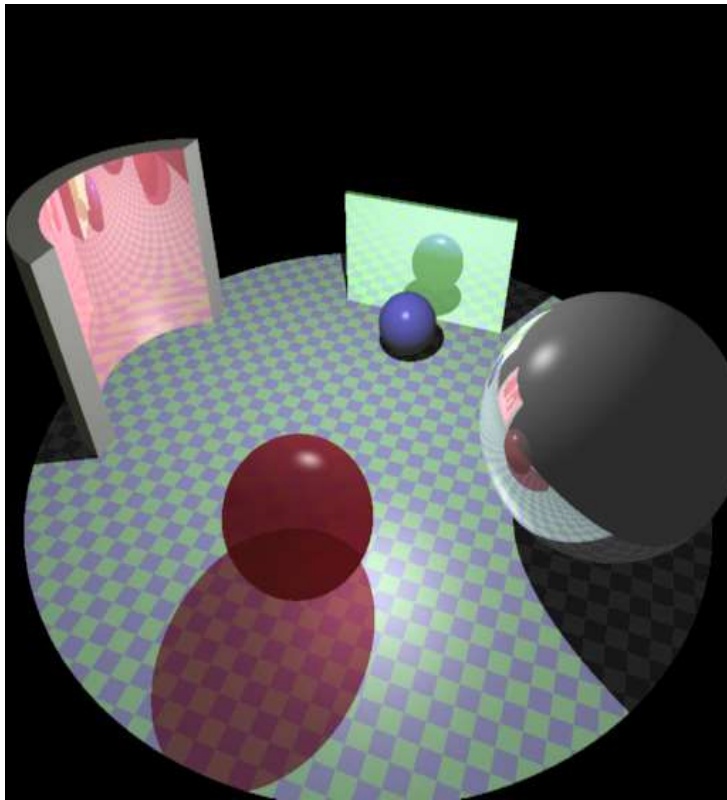
# Distributed ray-tracing



# Two-pass ray-tracing



# Classic vs Two-pass ray-tracing



# RAY TRACING CLÀSSIC

# Color d'un punt P

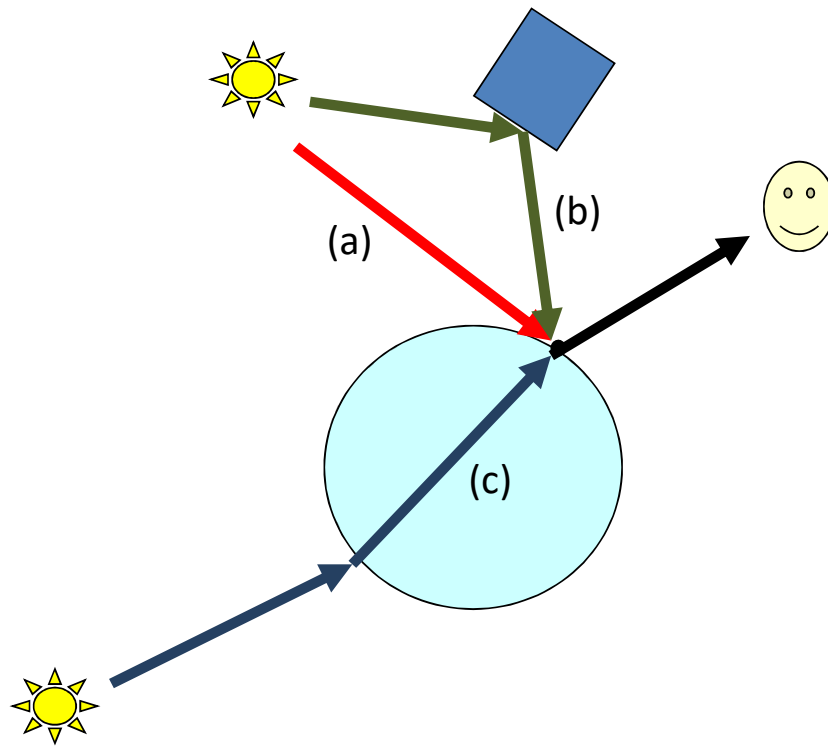
- El color amb que un determinat observador veu un cert punt P es calcula:

$$I(P) = I_D(P) + I_R(P) + I_T(P)$$

- $I_D(P)$  és color degut a la **llum directa** dels focus.
- $I_R(P)$  és color degut a la **llum indirecta que es reflecteix** a P en direcció cap a l'observador.
- $I_T(P)$  és color degut a la **llum indirecta que es transmet** des de P en direcció cap a l'observador.

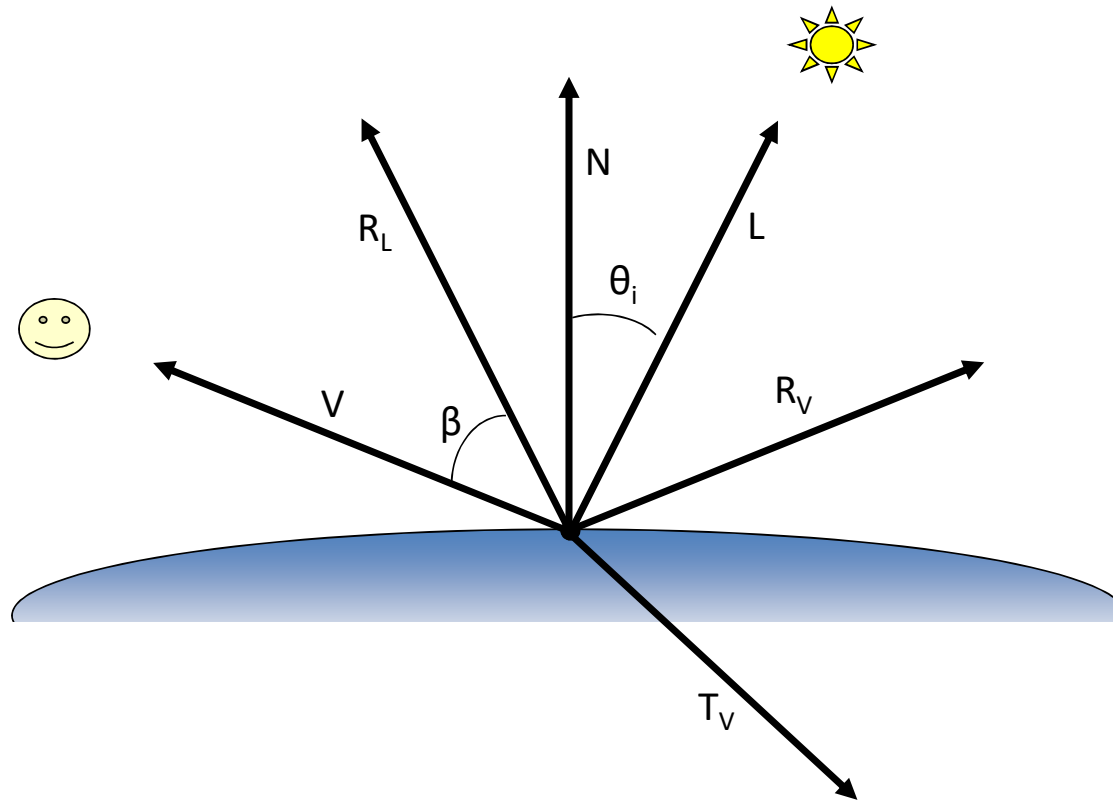
# Color d'un punt P

$$I(P) = I_D(P) + I_R(P) + I_T(P)$$





# Color d'un punt P: notació

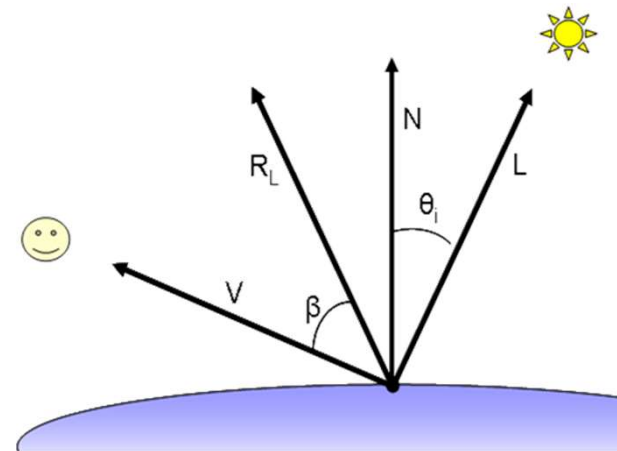


# Contribució Ilum directa $I_D$

$$I(P) = I_D(P) + I_R(P) + I_T(P)$$

$$I_D(P) = K_a I_a + K_d \sum I_L \cos(\theta_i) + K_s \sum I_L \cos^n(\beta)$$

- $\cos(\theta_i) = N \cdot L$
- $\cos(\beta) = R_L \cdot V$
- El sumatori només considera les fonts de llum no ocluides (ombres)

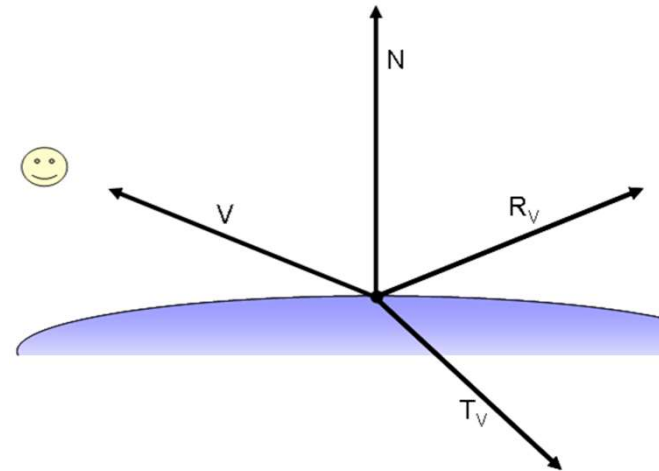


# Contribución llum indirecta $I_R$ , $I_T$

$$I(P) = I_D(P) + I_R(P) + I_T(P)$$

$$I_R(P) = K_R L_R$$

$$I_T(P) = K_T L_T$$



- $K_R$   $K_T$  coeficients empírics de reflexió/transmissió especular
- $L_R =$  llum que incideix en P en la direcció  $R_V$
- $L_T =$  llum que incideix en P en la direcció  $T_V$

Es calculen recursivament, traçant un nou raig reflectit i un altre transmès

# Algorisme

```
acció rayTracing
  per i en [0..w-1] fer
    per j en [0..h-1] fer
      raig:=raigPrimari(i, j, camera);
      color:=traçarRaig(raig, escena,  $\mu$ );
      setPixel(i, j, color);
    fper
  fper
faccio
```

# Algorisme

```
funció traçar_raig(raig, escena,  $\mu$ )
  si profunditat_correcta() llavors
    info:=calcula_interseccio(raig, escena)
    si info.hi_ha_interseccio() llavors
      color:=calcular_ID(info,escena); // ID
      si es_reflector(info.obj) llavors
        raigR:=calcula_raig_reflectit(info, raig)
        color+= KR*traçar_raig(raigR, escena,  $\mu$ ) //IR
      fsi
      si es_transparent(info.obj) llavors
        raigT:=calcula_raig_transmès(info, raig,  $\mu$ )
        color+= KT*traçar_raig(raigT, escena, info. $\mu$ ) //IT
      fsi
    sino color:=colorDeFons
  fsi
sino color:=Color(0,0,0); // o colorDeFons
fsi
retorna color
ffunció
```

# Shader (FS)

```
void main() {  
    vec3 obs = gl_ModelViewMatrixInverse[3].xyz;  
    vec3 dir = normalize(pos - obs);  
    Ray ray = Ray(obs,dir);  
    gl_FragColor = trace(ray, 1.0);  
}
```

# Shader (FS)

```
vec4 trace(Ray ray, float mu) {
  if ( intersectScene(ray, Phit, Nhit,Kd, Kr, Kt) ) {
    Ray shadowRay = Ray(Phit, normalize(lightPos-Phit));
    bool inShadow = intersectScene(shadowRay, aux, aux,aux4, aux4, aux4);
    if (inShadow) shadow = 0.2; else shadow = 1.0;
    color+= shadow*light(Nhit, -ray.dir, lightPos-Phit, Kd, vec4(1.0));
    vec3 R = reflect(ray.dir, Nhit);
    color += Kr*trace1(Ray(Phit, R), mu);
    if (mu==1.0) muHit=1.5;
    else { muHit = 1.0; Nhit*=-1.0;}
    vec3 T = refract(ray.dir, Nhit, mu/muHit);
    if (length(T)>0.0) color+=Kt*trace1(Ray(Phit, T), muHit);
  }
  else color+=samplePanorama(ray.dir);
  return color;
}
```

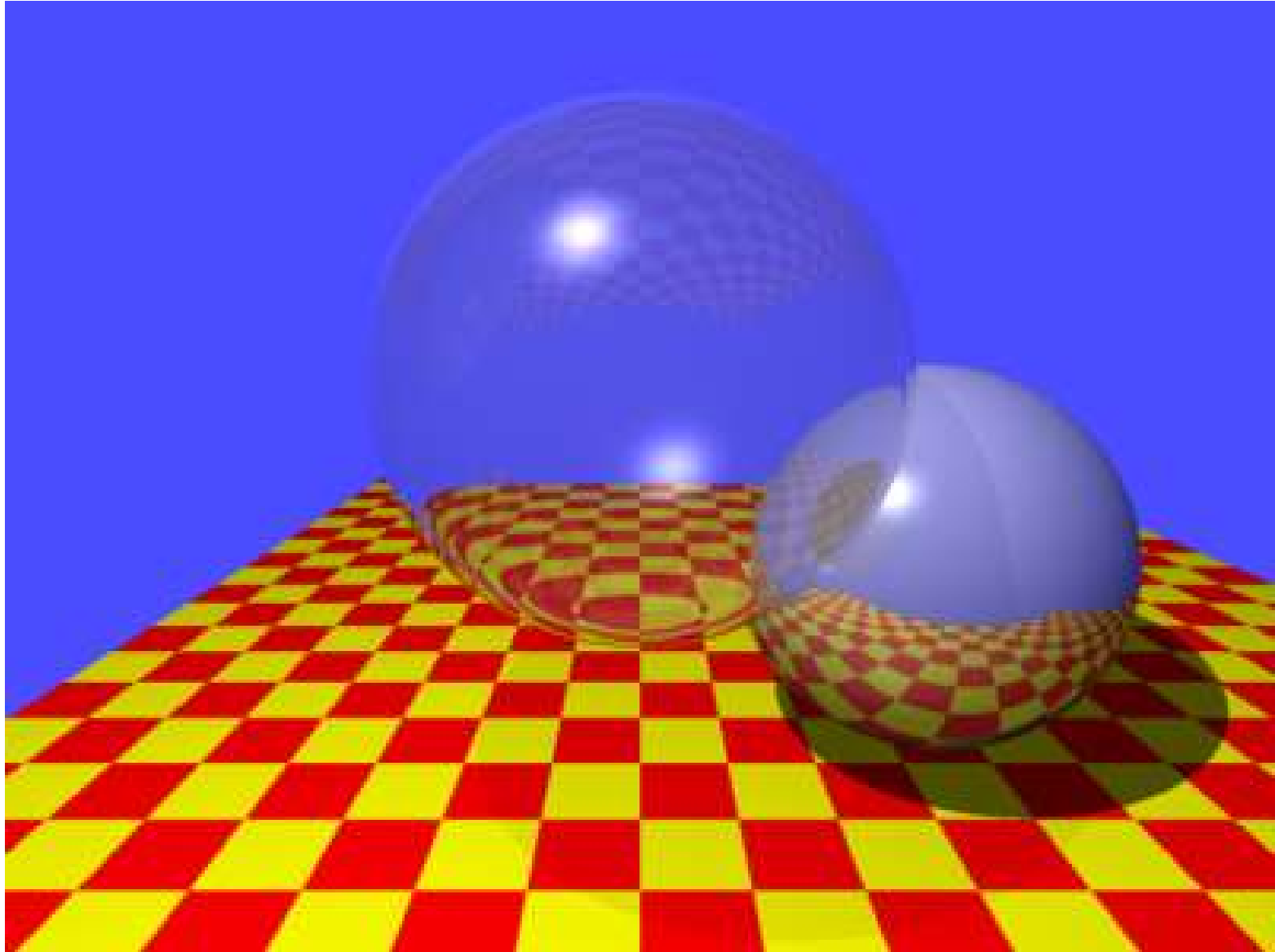
# Shader (FS)

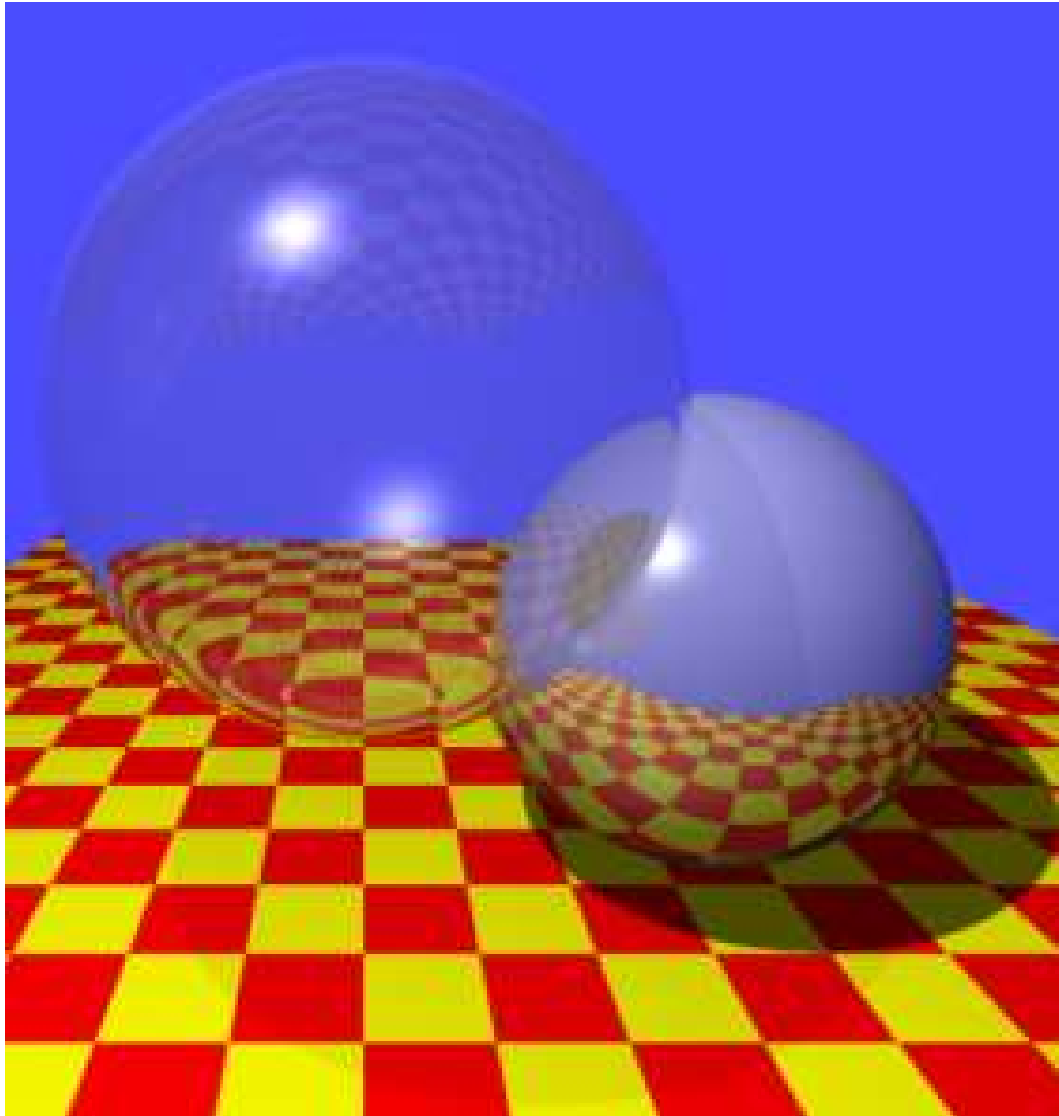
```
vec4 samplePanorama(vec3 R)
{
    float psi = asin(R.y);
    float theta = atan(R.z, R.x);
    float t = 1.0 - psi/PI + 0.5;
    float s = theta/(2.0*PI) + 0.7;
    return texture(panorama,vec2(s,t));
}
```



# Shader (FS)

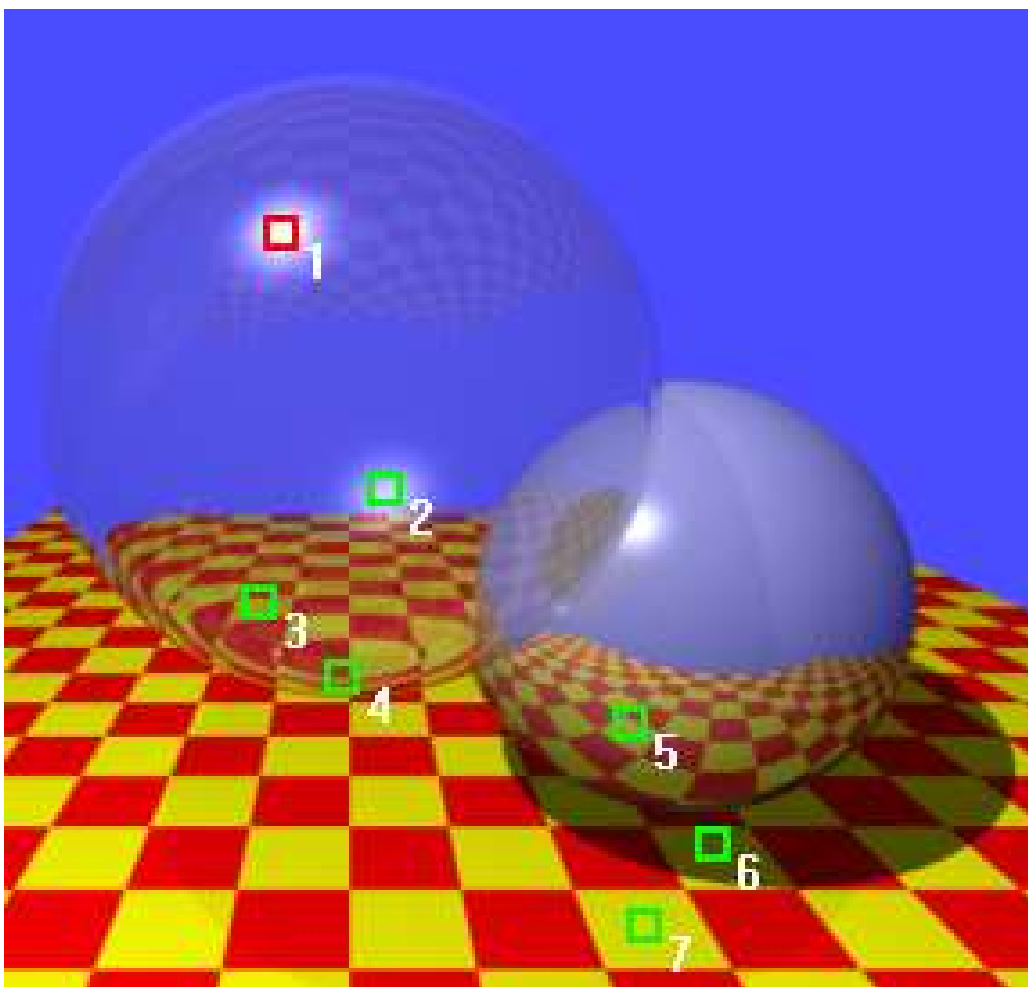
```
vec4 samplePanorama(vec3 R)
{
    float psi = asin(R.y);
    float theta = atan(R.z, R.x);
    float t = 1.0 - psi/PI + 0.5;
    float s = theta/(2.0*PI) + 0.7;
    return texture(panorama, vec2(s,t));
}
```





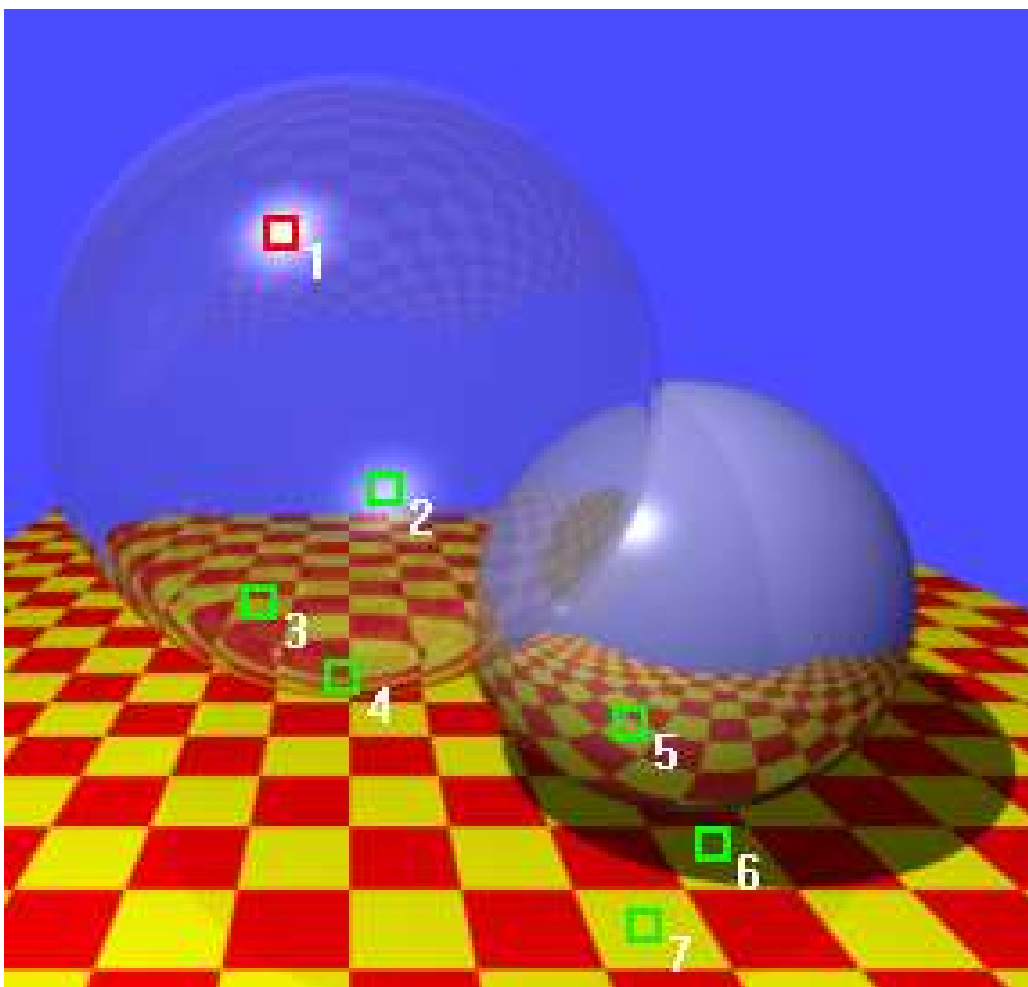
Escena:

- Esfera **buida** transparent.
- Esfera opaca blanca mirall.
- Tauler escacs
- Fons blau
- Llum blanca



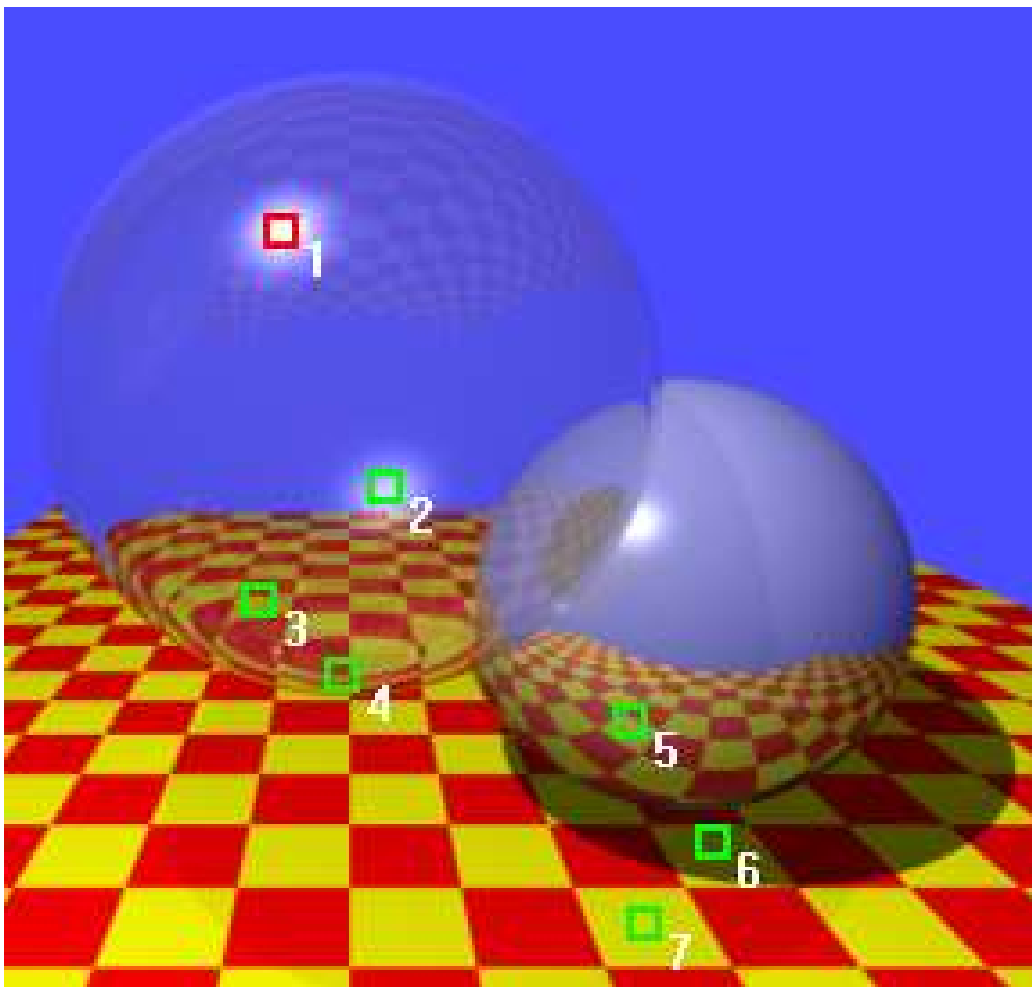
### Raig 1

- El raig primari gairebé coincideix amb la reflexió del vector  $L$ : podem veure una taca especular (specular highlight).
- La contribució principal és la component especular de  $I_D(P)$ .



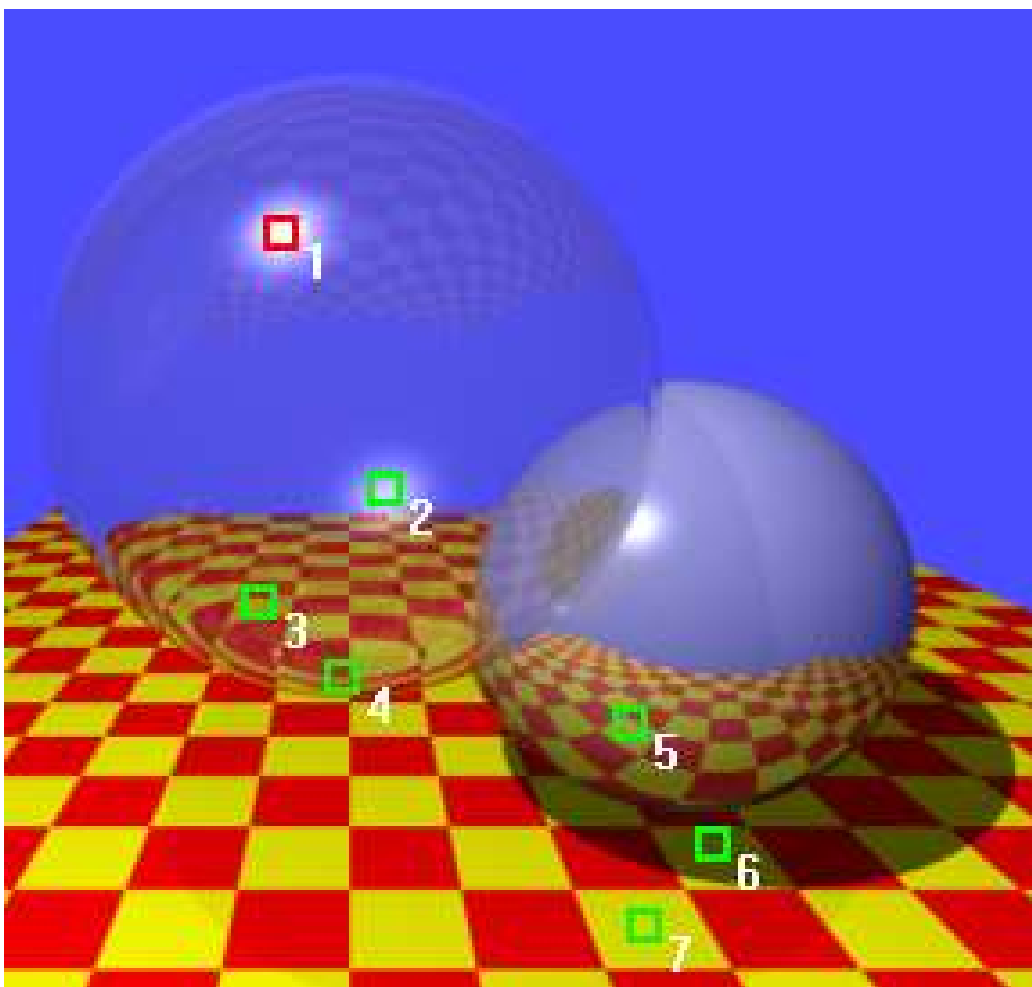
## Raig 2

- És la mateixa situació que el raig 1, però ara la taca especular apareix a la paret interna de l'esfera (que és buida).
- Aquest raig mostra un error acceptat en ray-tracing: el raig de la font de llum travessa l'esfera sense refracció (només comparem  $L$ ,  $N$  ignorant el fet que som a dins de l'esfera). Per tant, la taca especular està en una posició errònia, però no tenim cap intuïció de la posició correcta.



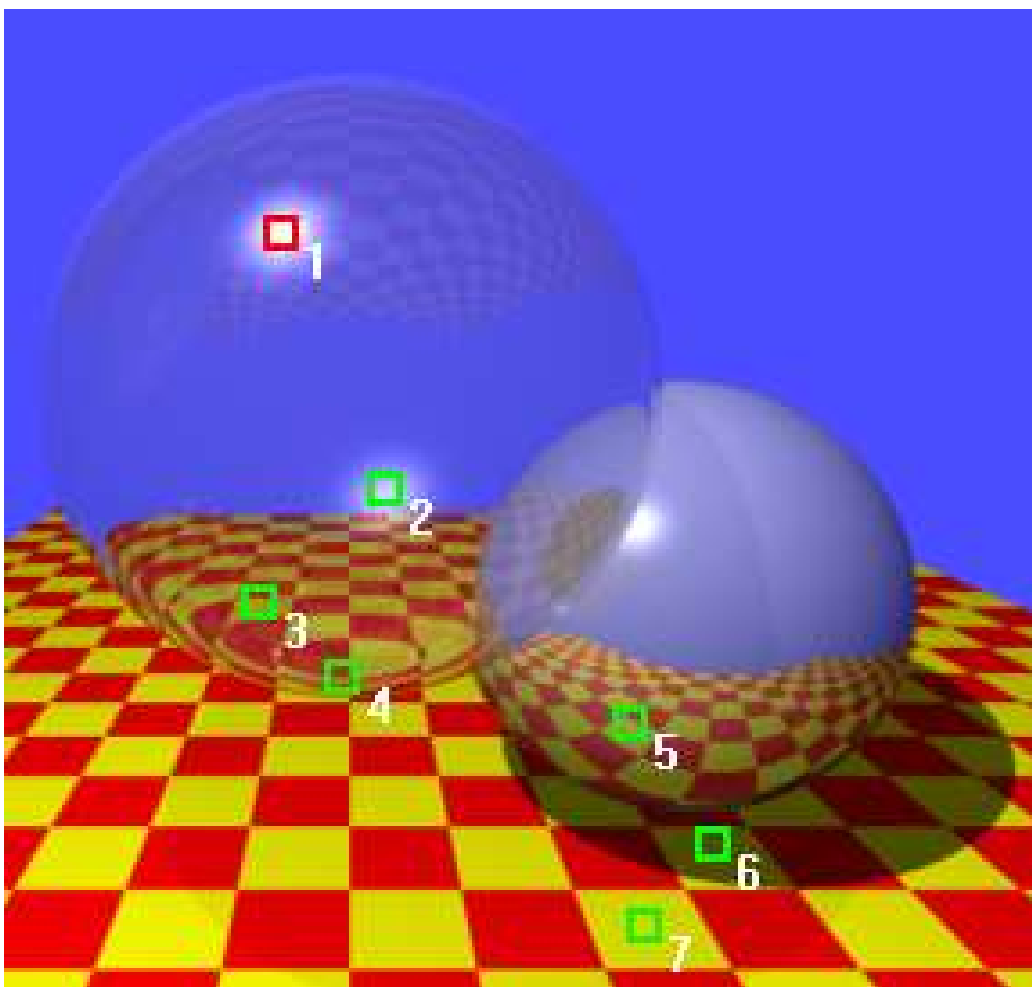
### Raig 3

- En totes les interseccions d'aquest raig amb l'esfera la contribució local és nula.
- La contribució dominant és el tauler vermell-groc.
- Hi ha una lleugera distorsió deguda a la refracció.
- Hi ha una barreja de dos taulers: el transmès i el reflectit a la superfície exterior de l'esfera.



#### Raig 4

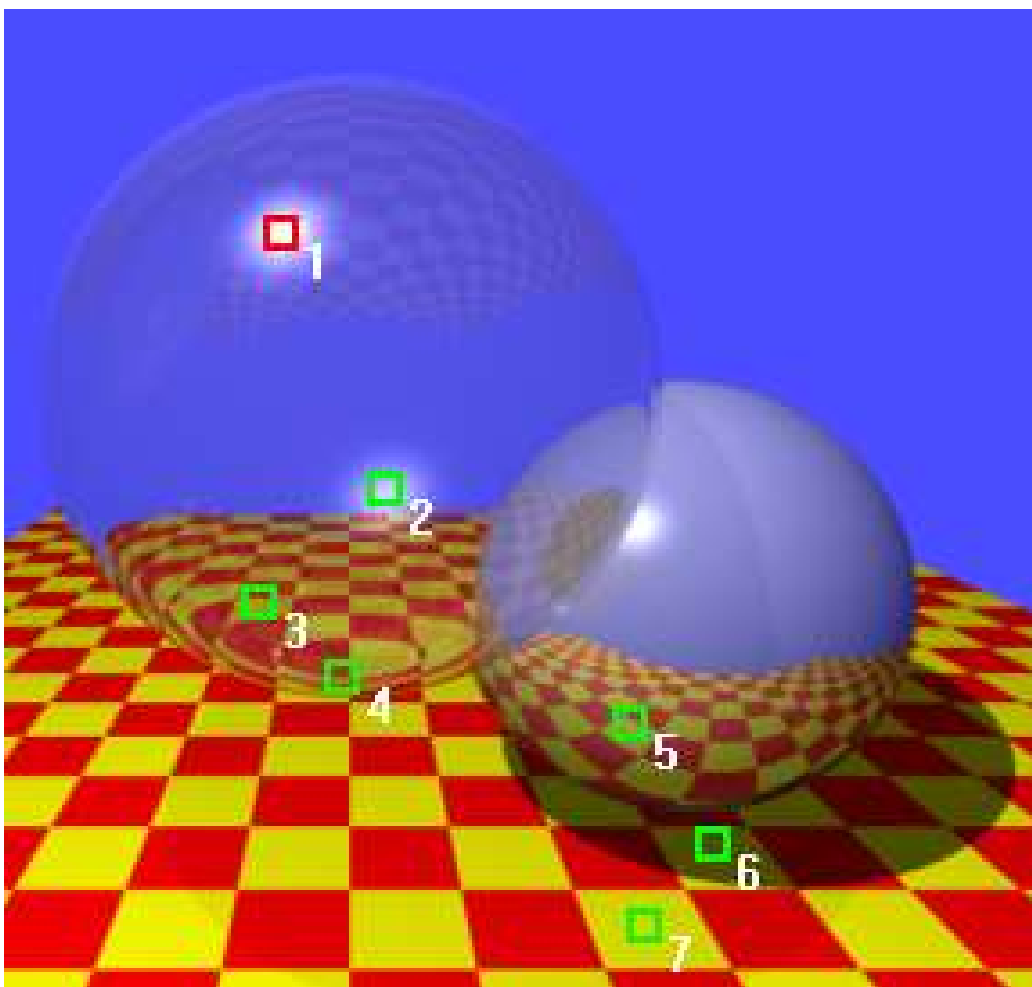
- Igual que el raig anterior, però ara la distància recorreguda dins el cristall és més significativa, amb la qual cosa l'efecte de la refracció és més notori.



### Raig 5

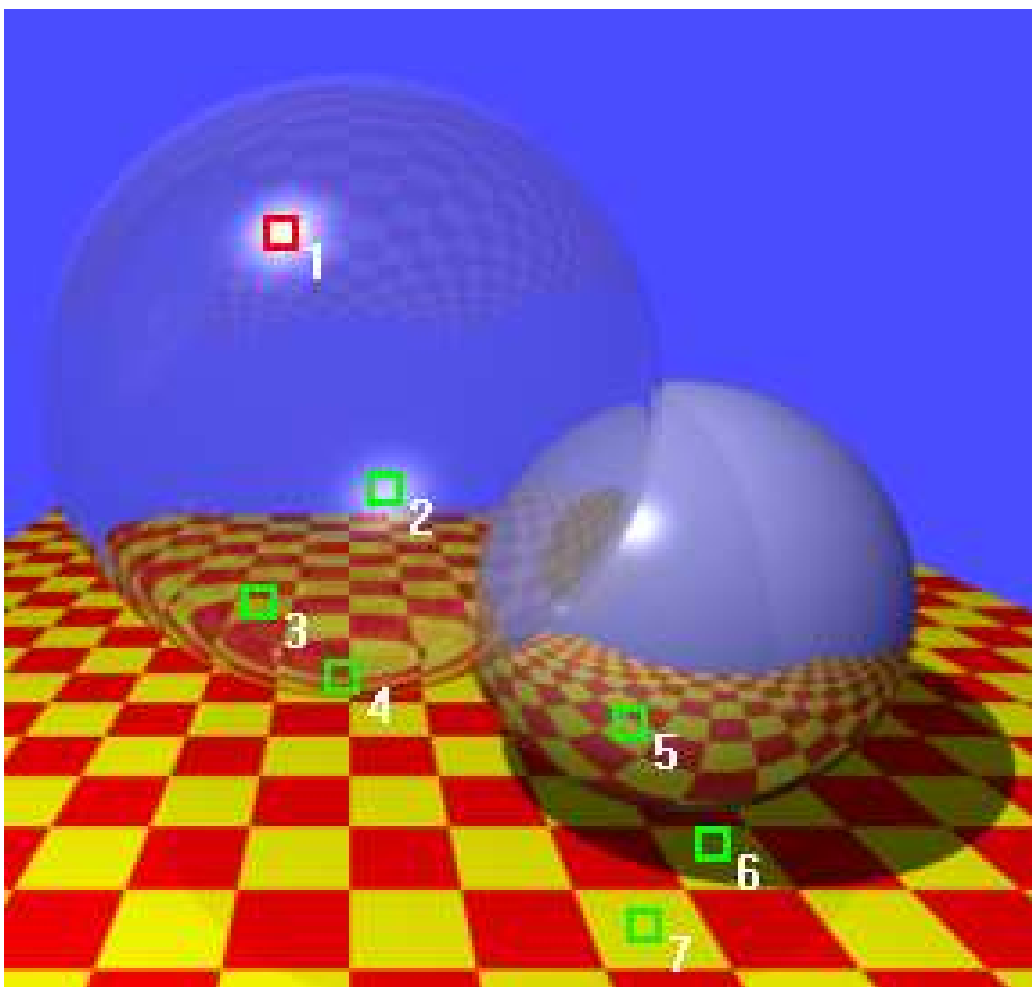
- El color resultant és la barreja del tauler (raig reflectit) amb el blanc de l'esfera (lleugerament difosa).





### Raig 6

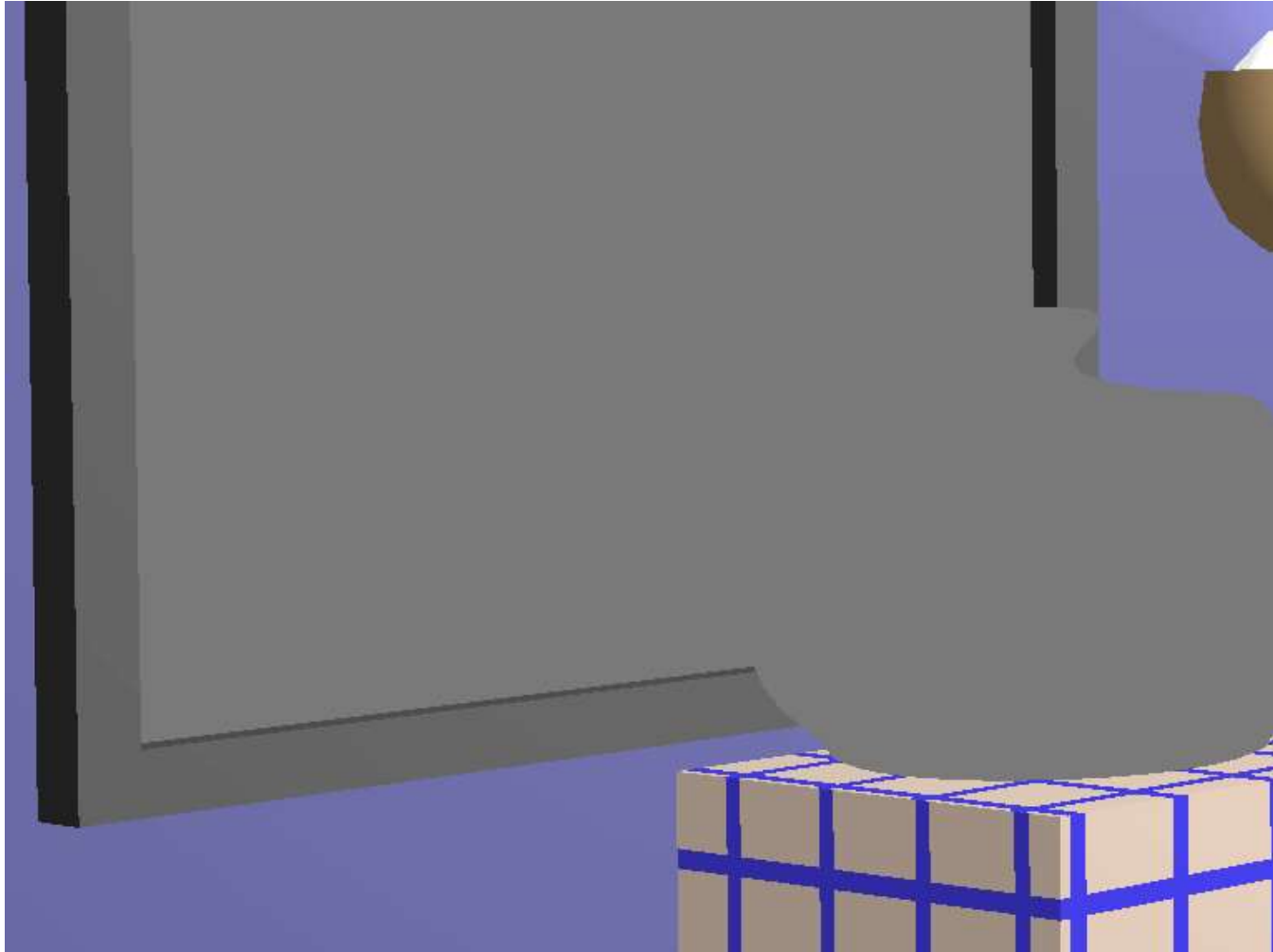
- El raig intersecta el tauler; el punt d'intersecció està a l'ombra (el shadow ray intersecta l'esfera opaca).



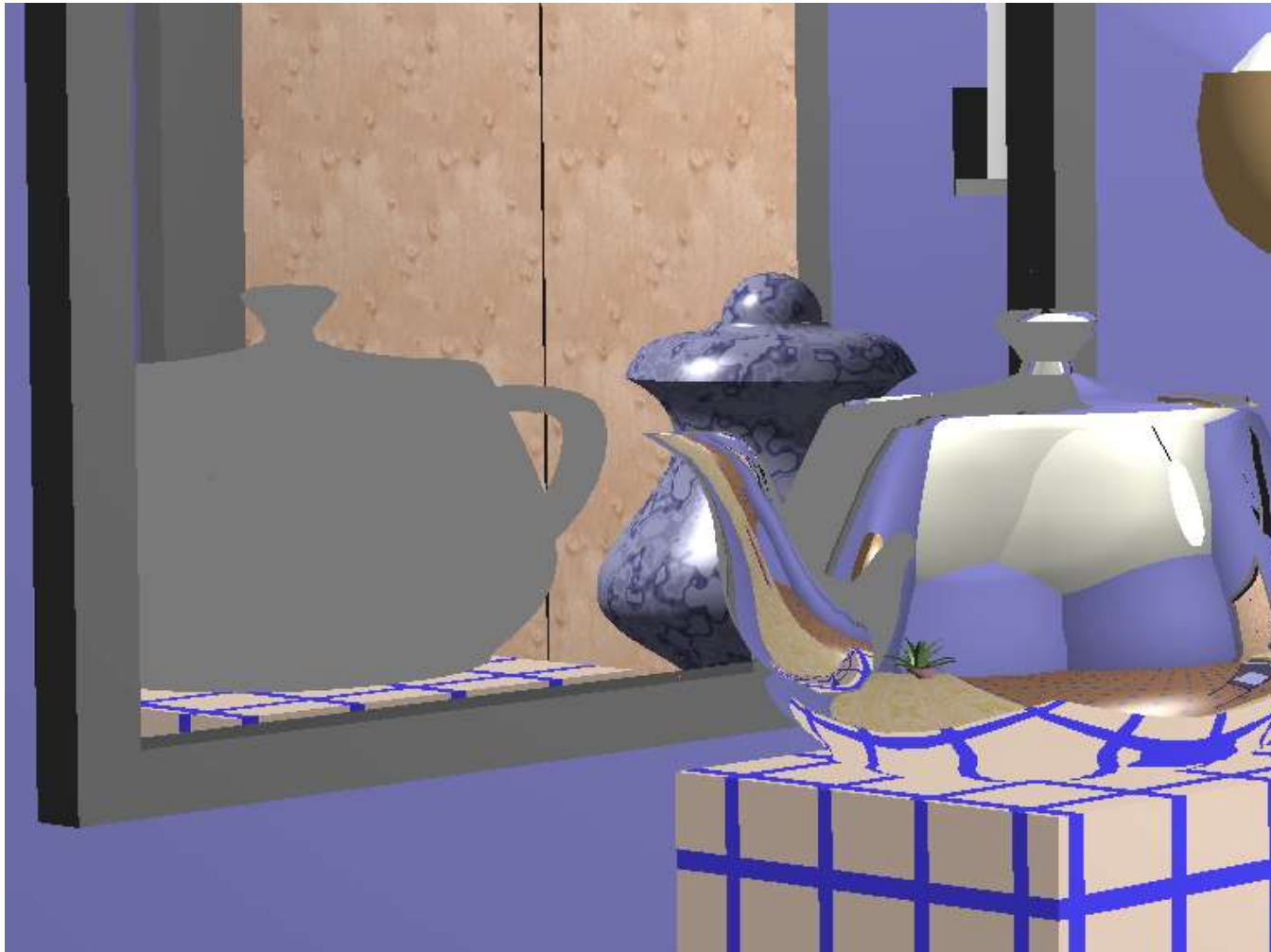
### Raig 7

- Exactament igual que abans però ara el shadow ray intersecta l'esfera transparent i per tant gairebé no reduïm la contribució de la font de llum.
- Novament, no estem tenint en compte que la llum es refractaria a l'esfera i per tant el contorn de l'ombra no està a la posició correcta.

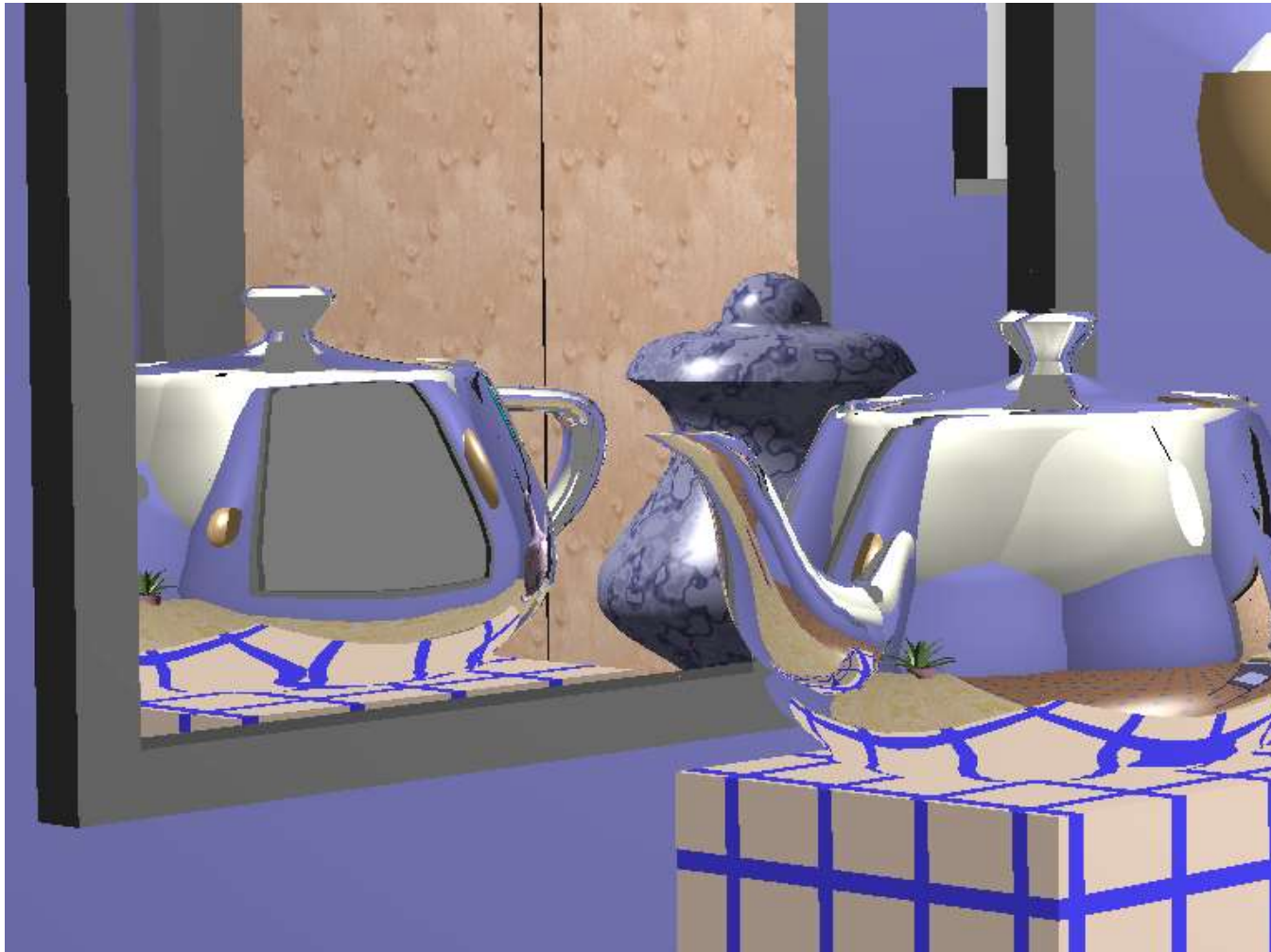
# Ray-tracing clàssic. Profunditat 0



# Ray-tracing clàssic. Profunditat 1



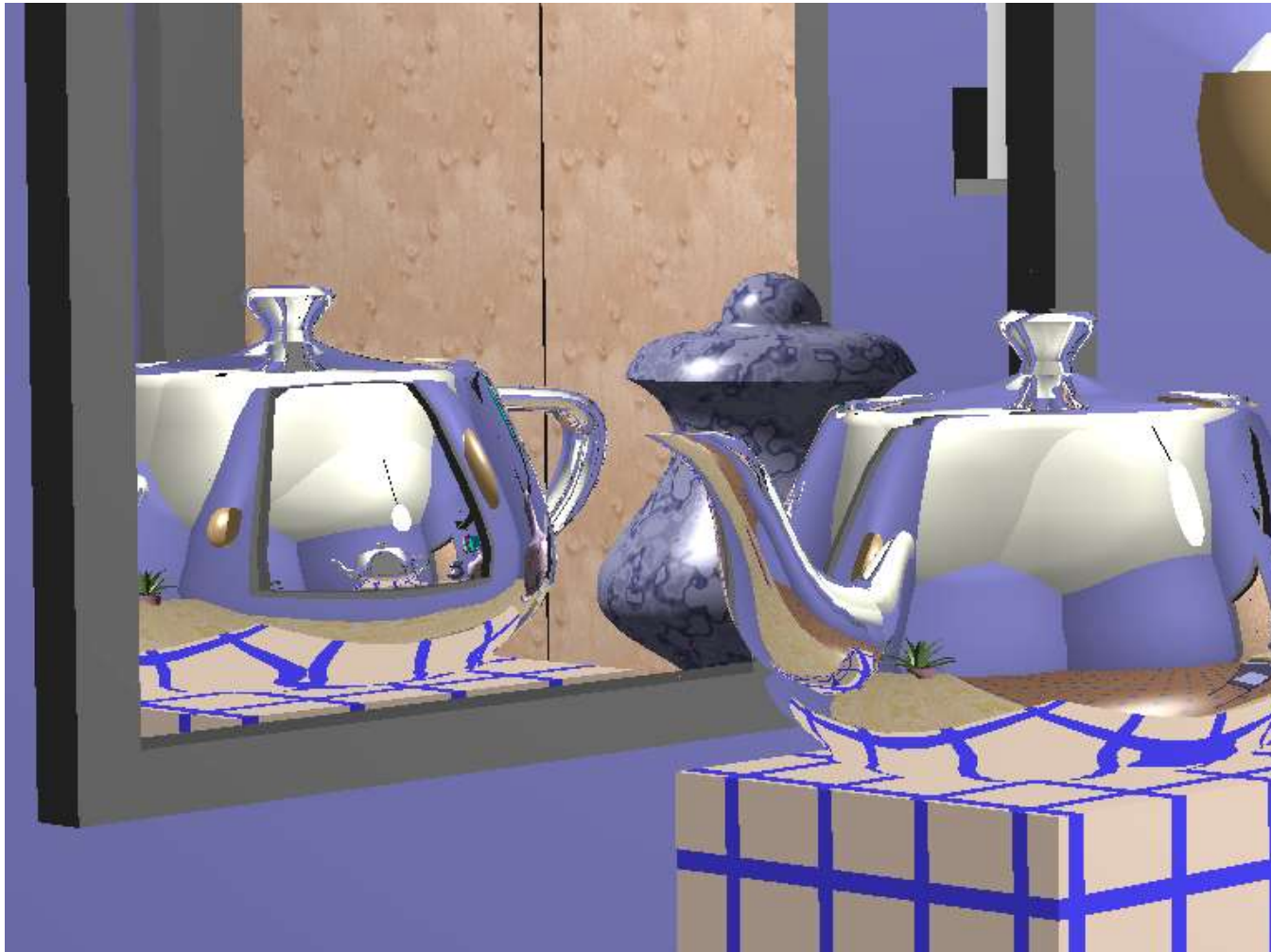
## Ray-tracing clàssic. Profunditat 2



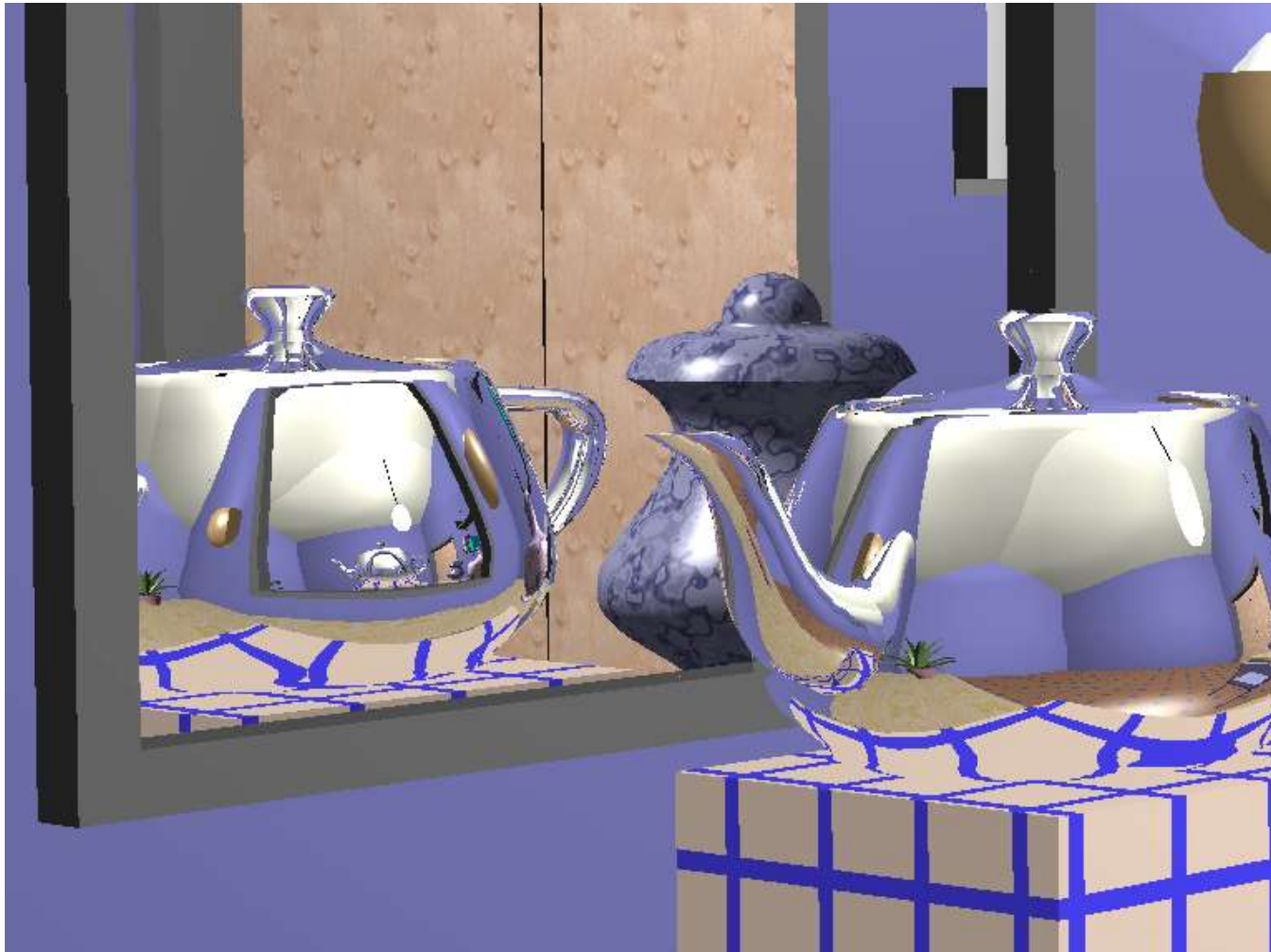
## Ray-tracing clàssic. Profunditat 3



## Ray-tracing clàssic. Profunditat 4



## Ray-tracing clàssic. Profunditat 6





## Ray-tracing clàssic. Profunditat 8



## Ray-tracing clàssic. P. 8 + AA



# Contribució local i global

