

RAYTRACER



Amaury Patard, Ahmed Bougacha
Mickael Rodrigues, Aymeric Beaumet

XML
GRAPHICSMANAGER

CAMERAS
SAMPLING

MATERIAUX RENDERPASS

OBJETS
MESHES

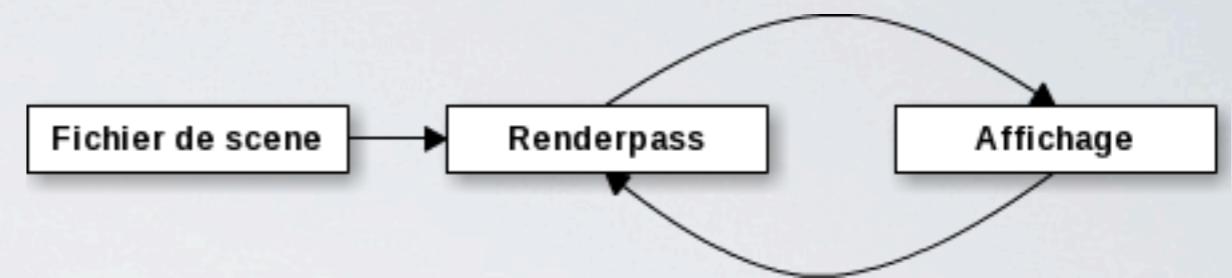
FONCTIONNEMENT PARALLELISATION

RENDUS

PRESENTATION

PRESENTATION

raytracer en OOC



permet un rendu rapide, basé sur le concept de *renderpass*

permet aussi le *pathtracing*

XML

fichier de scène

XML

Usage: ./rt scene.xml

```
<scene>
  <image>
    <camera type="...." />
    <rendering size="..." samples="..." />
  </image>
  <renderpass>
    <pass type="..." />
  </renderpass>
  <materials>
    <material type="..." id="..." />
  </materials>
  <objects>
    <object type="..." id_mat="..." coor="..." rot="..." />
  </objects>
  <lights>
    <light type="..." radiance="..."/>
  </lights>
</scene>
```

GRAPHICSMANAGER

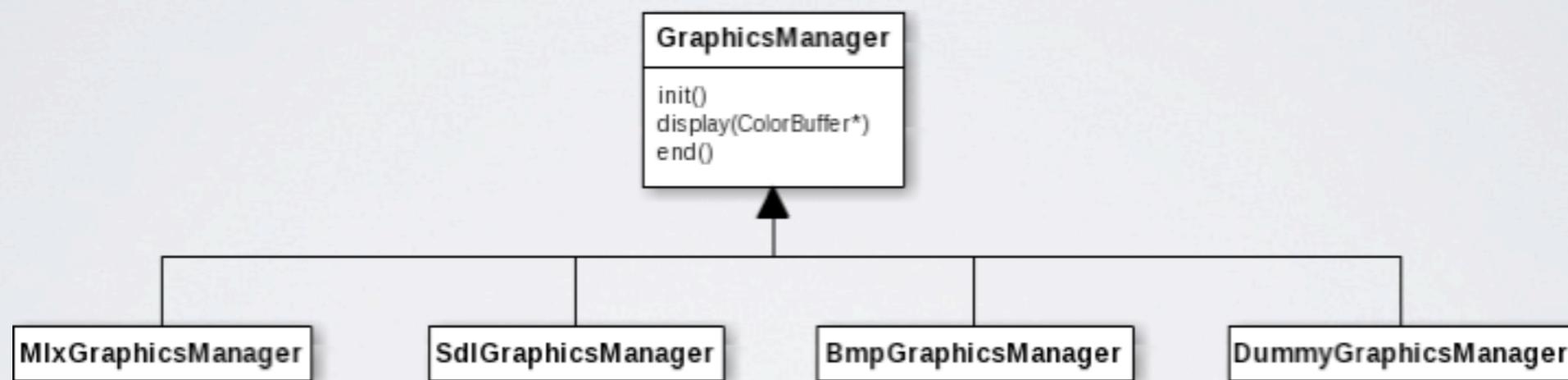
GRAPHICS MANAGER

Usage : ./rt scene.xml Display:[mlx|sdl|bmp|none]

La sortie se fait grâce au GraphicsManager

Il est très facile d'en ajouter (par exemple: ascii, png, GTK+, etc...)

GRAPHICS MANAGER



GRAPHICSMANAGER

- init(): initialise la mlx, la sdl ou autre
- display(): appelée après chaque renderpass afin de mettre à jour l'image rendue
- end(): préviens le manager qu'il n'y aura plus de mise-à-jour; (mlx: appel de mlx_loop, bmp: sauvegarde de l'image)

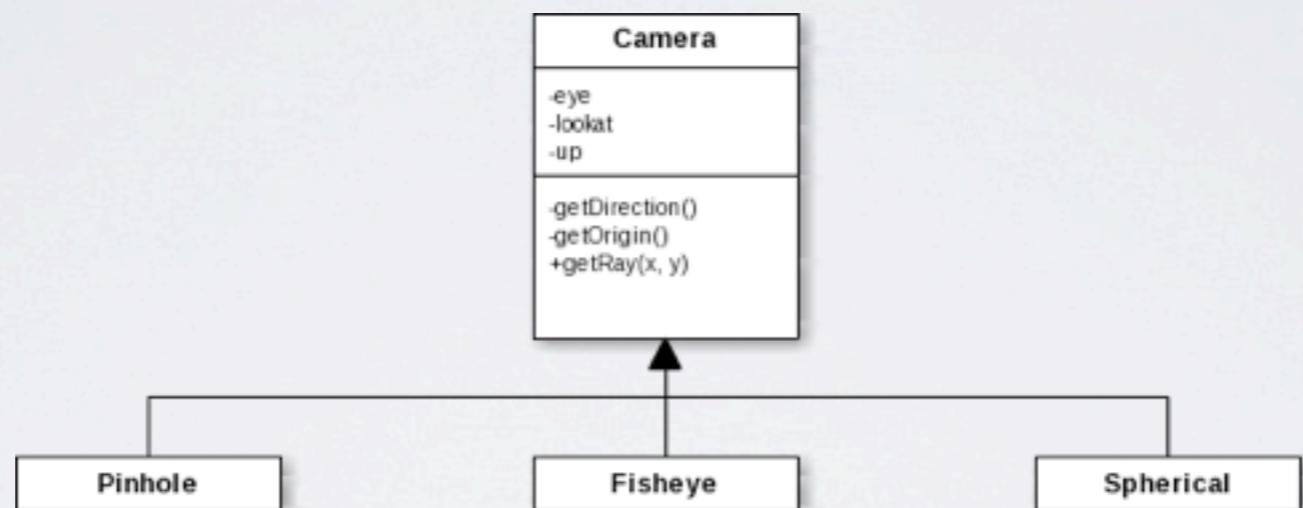
Dummy Graphics Manager

- Lorsque le rt est compilé avec NO_GRAPHICS (ne nécessite pas de serveur X)
- Lorsqu'un client ne veut pas d'affichage

CAMERAS

"A raytracer is never really good unless the camera is an eye in the head of a poet." - ORSON WELLES

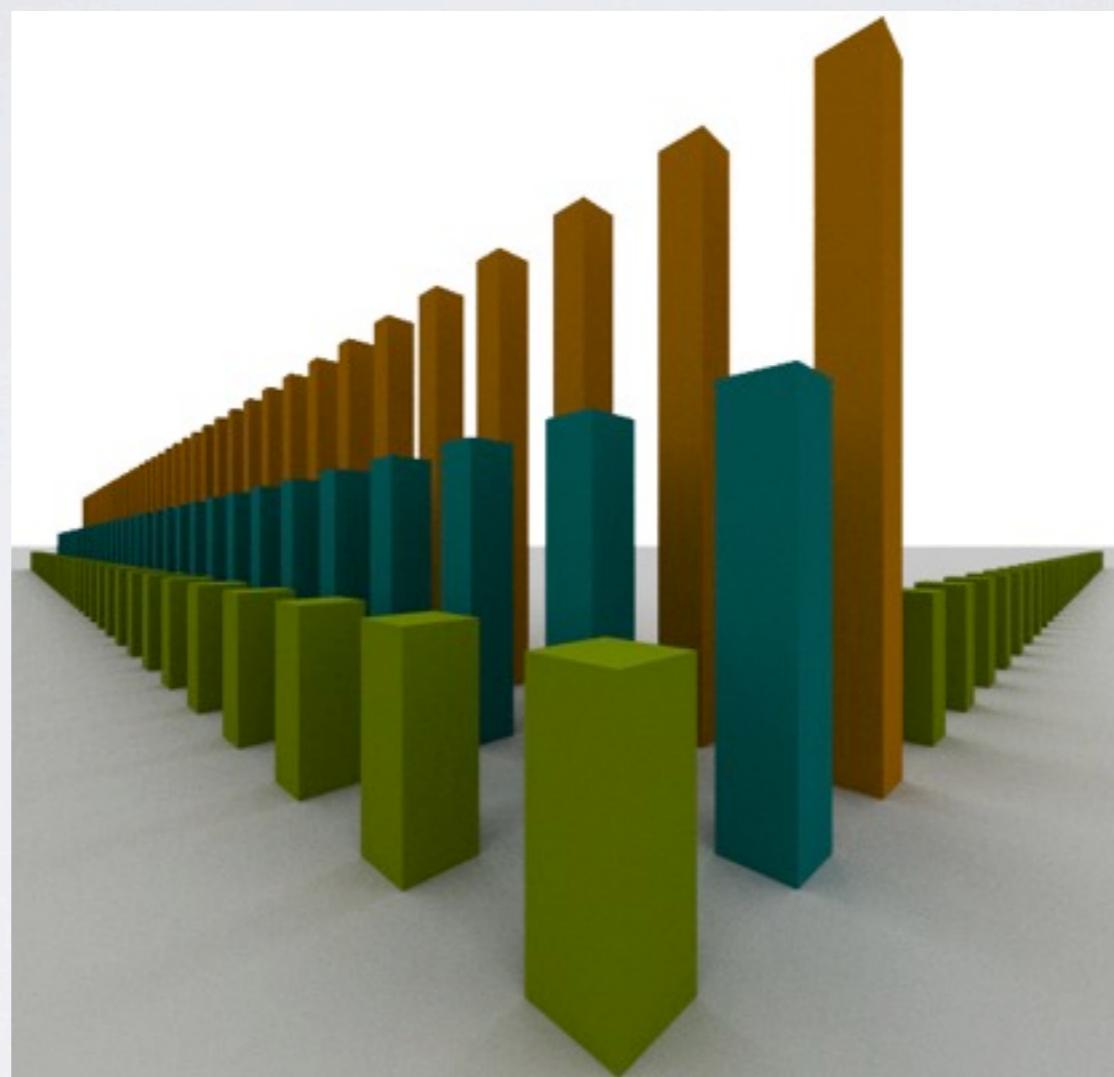
CAMERAS



CAMERAS

pinhole

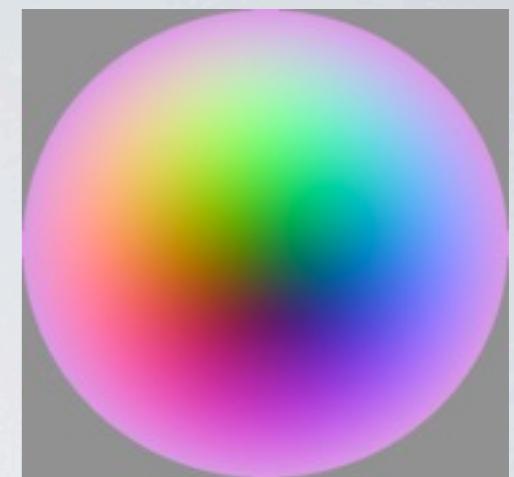
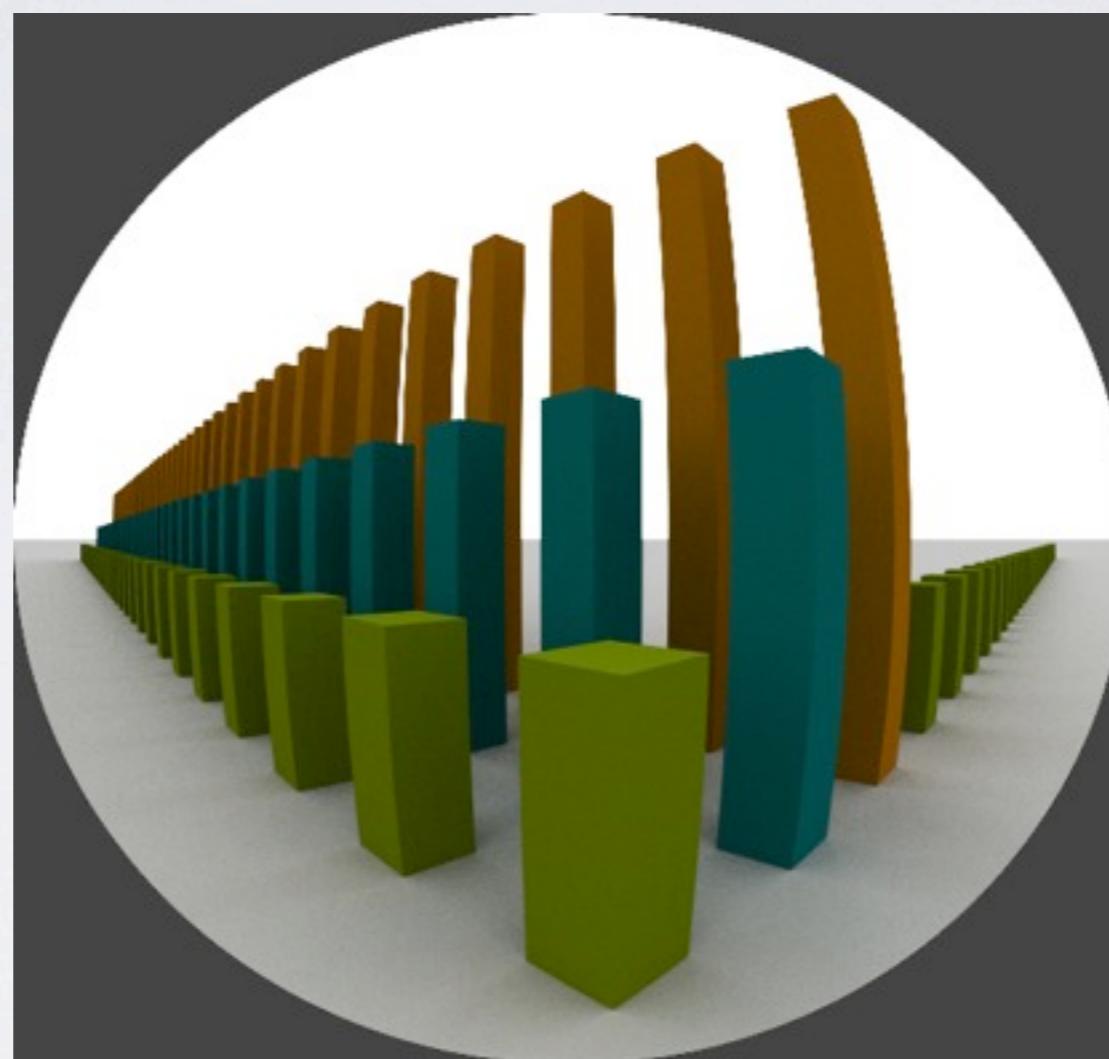
PINHOLE



CAMERAS

fisheye

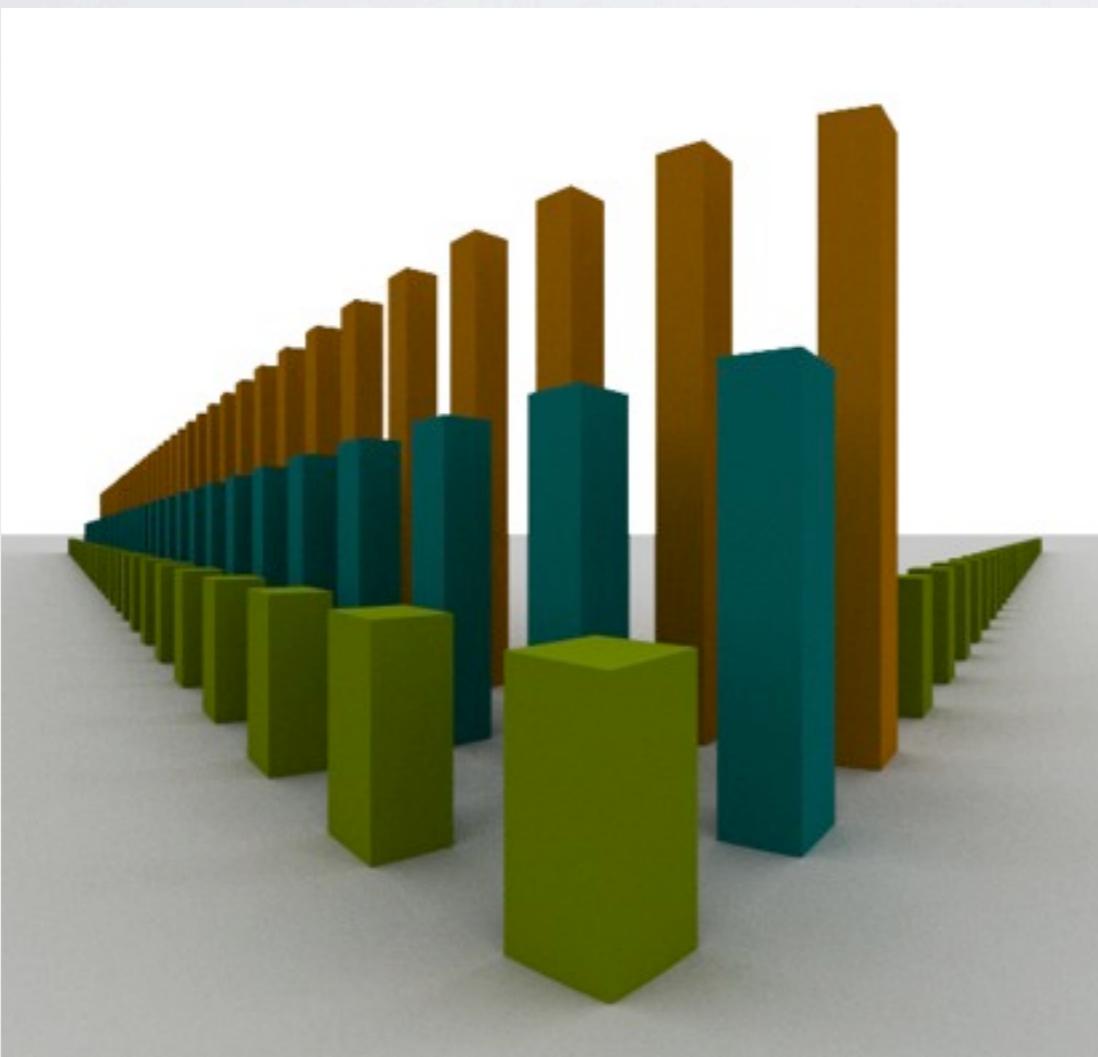
FISHEYE



CAMERAS

spherical

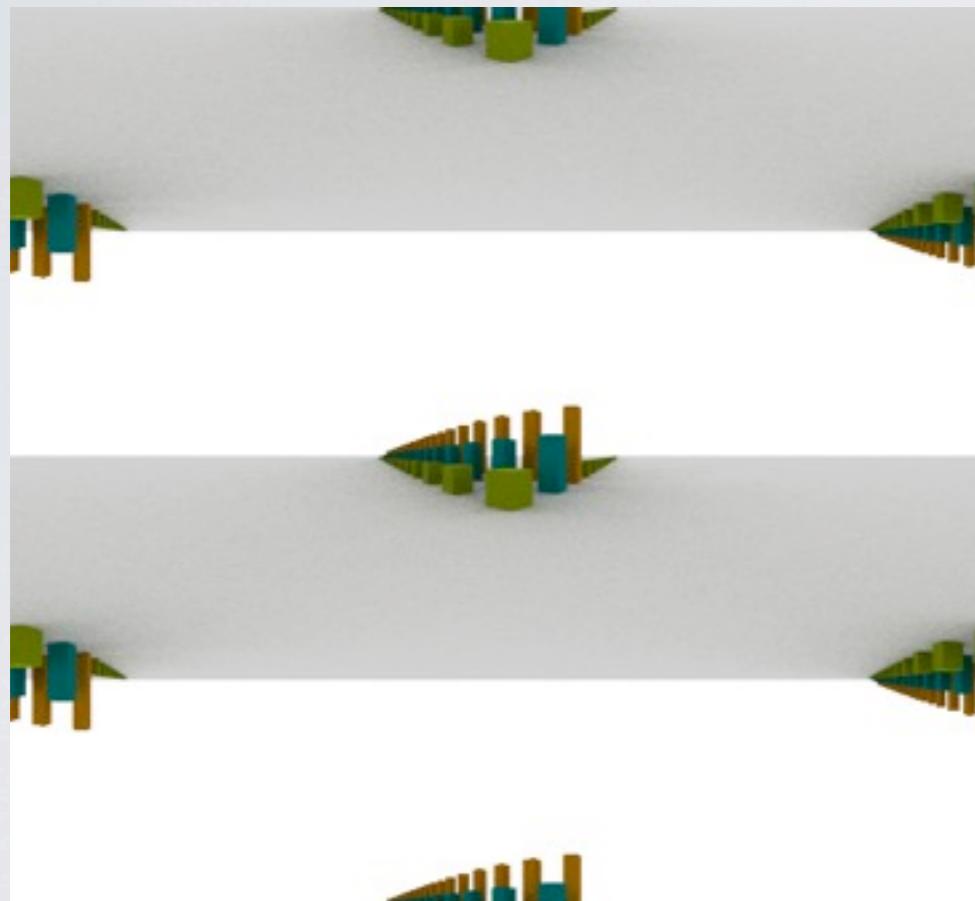
SPHERICAL



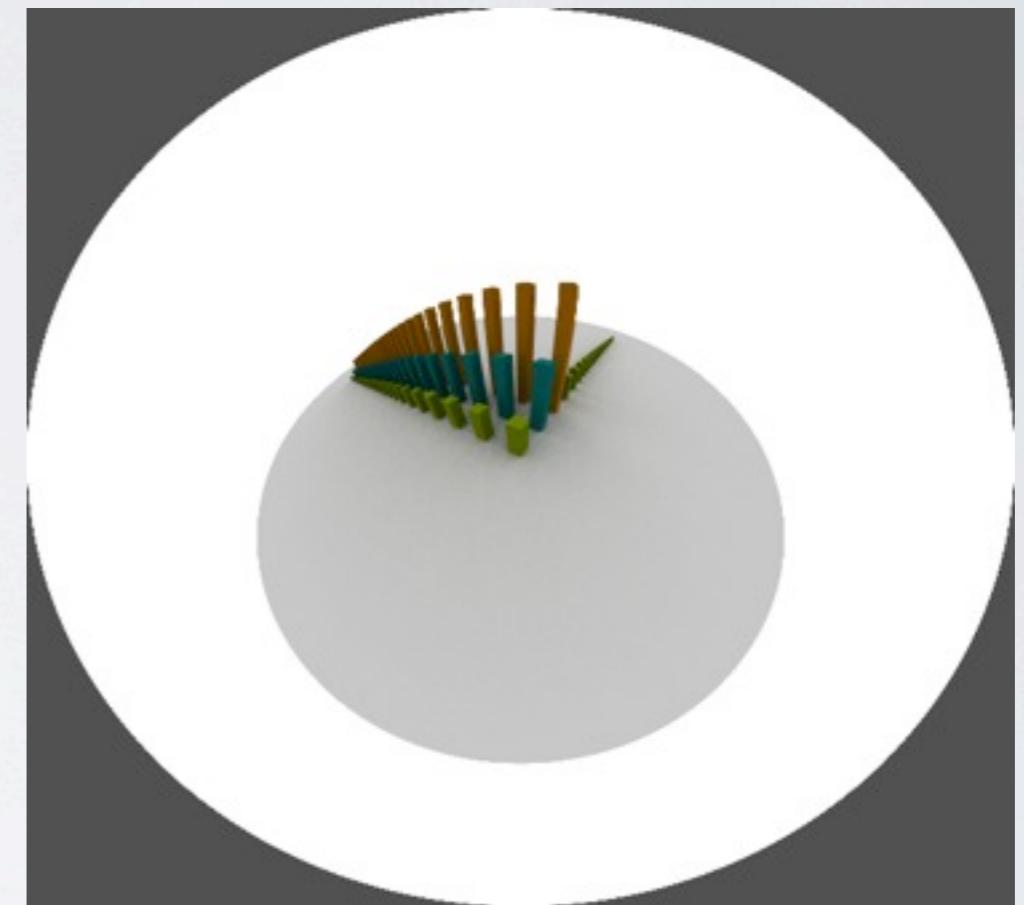
CAMERAS

grand angles

GRAND ANGLES - SPHERICAL / FISHEYE



Spherical @ 180x180 °



Fisheye @ 360 °

SAMPLING

"Yes, sampling has changed not just my way of raytracing and composing" - KLAUS SCHULZE

SAMPLING

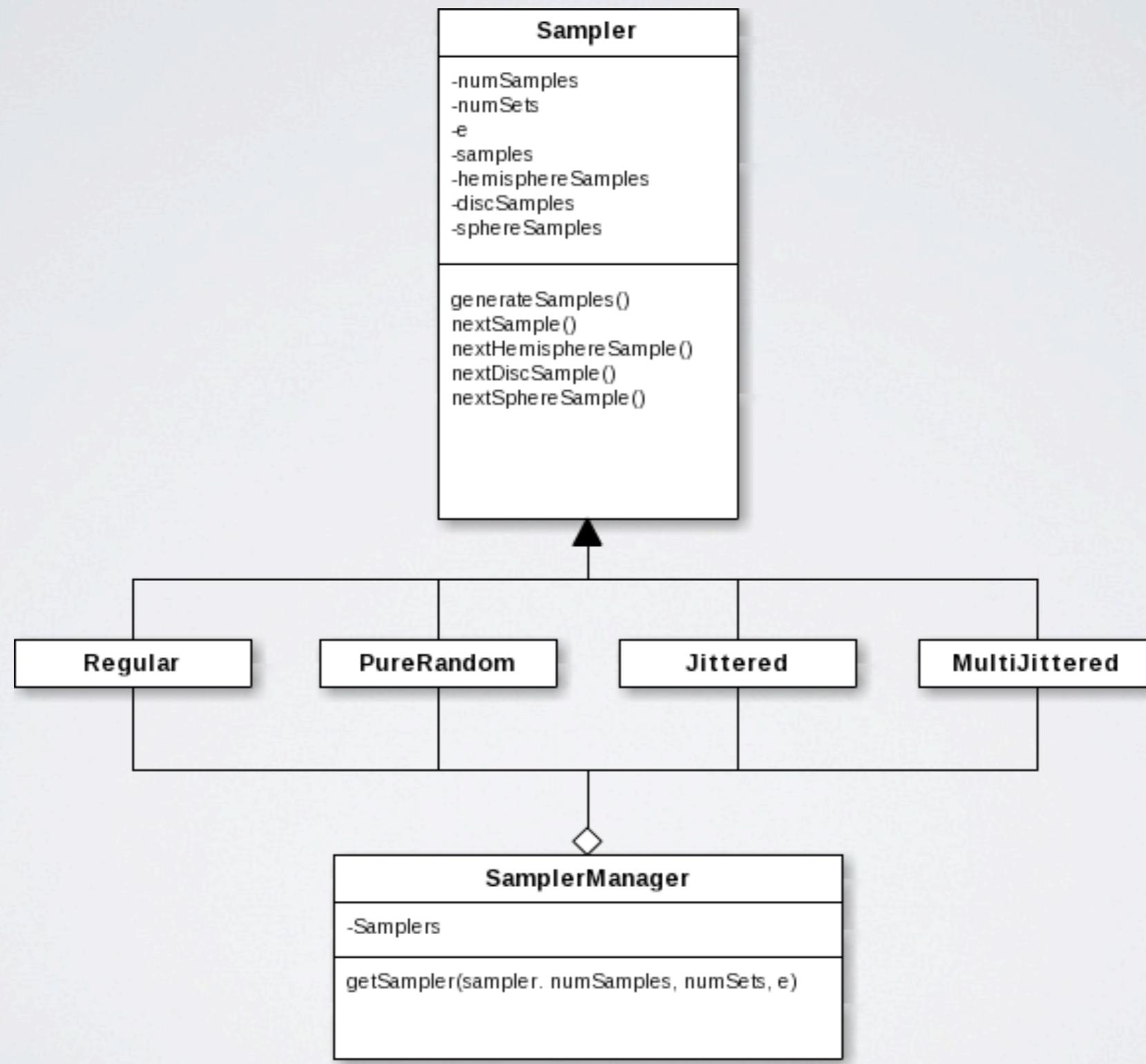
répartir des points dans un carré unitaire

les mapper sur une sphère - ou un hemisphere - unitaire

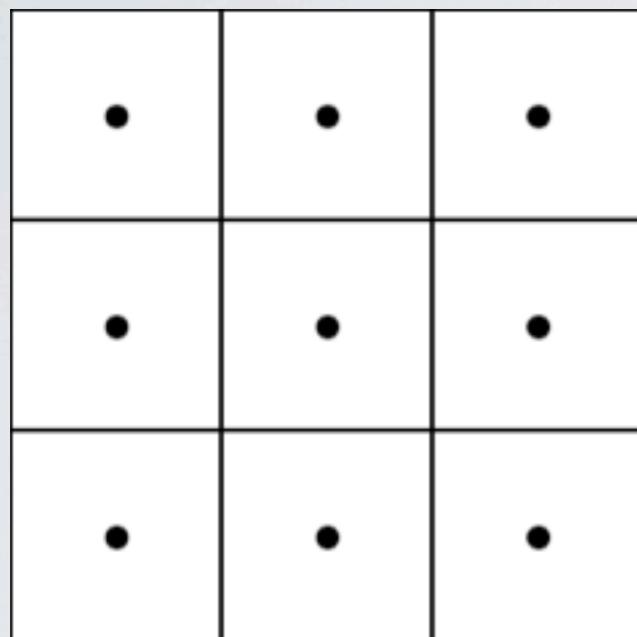
Les points doivent être:

- uniformément distribués en 2D (mais pas régulièrement)
- uniformément répartis sur les axes x/y
- assez espacés

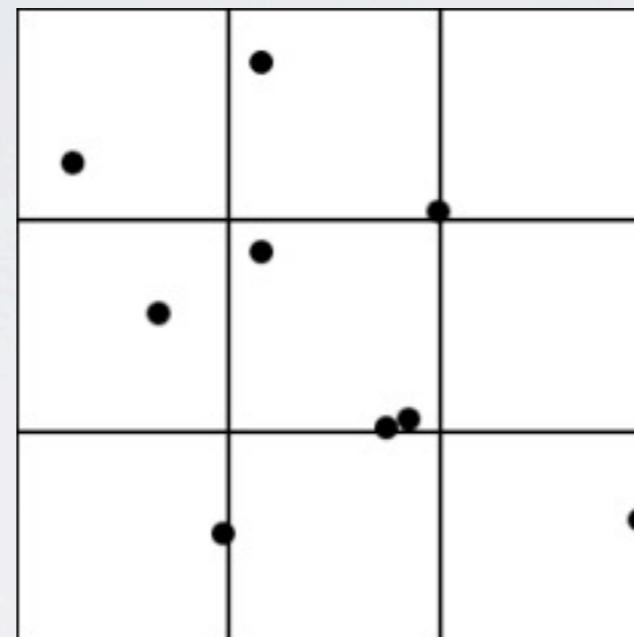
SAMPLING



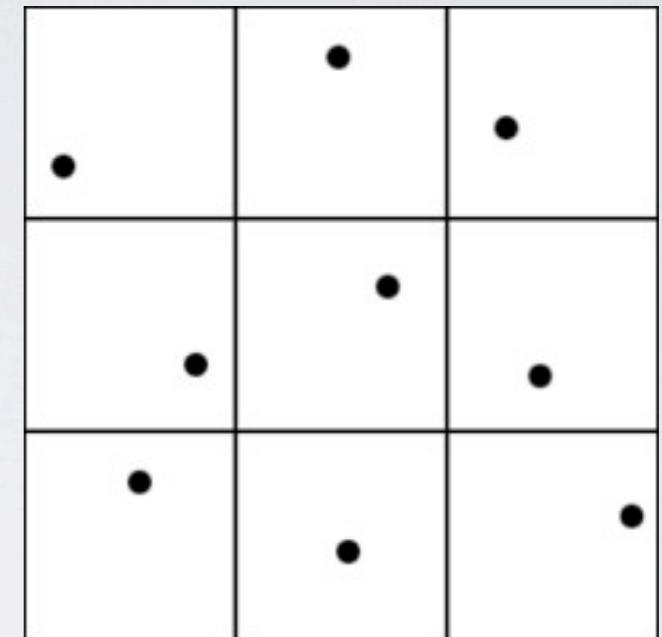
SAMPLING



regular



random



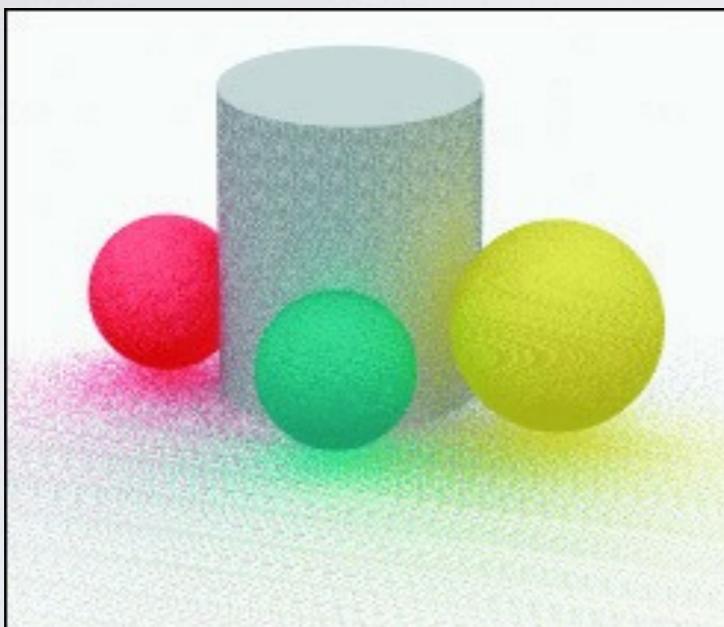
jittered

De nombreuses techniques sont mises en place pour éviter les *patterns*:

- utilisation d'un nombre premier de sets de samples
- mélange dans l'ordre d'utilisation des sets
- utilisation de samplers différents pour les différentes entités
- ...

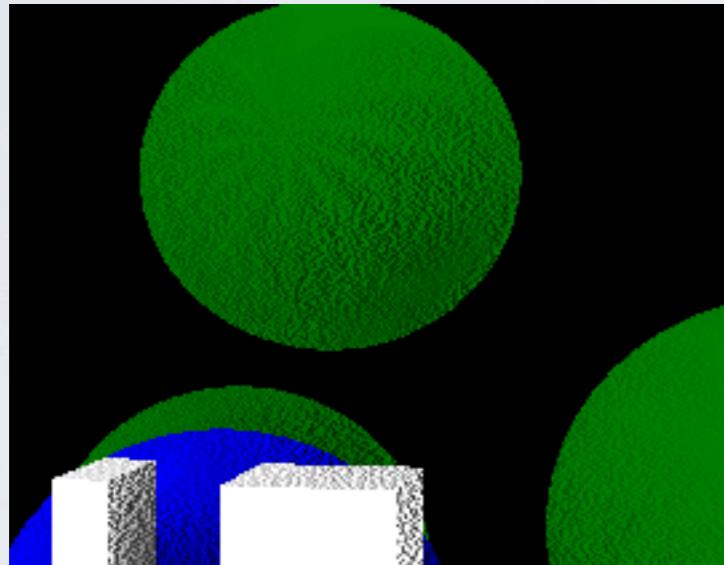
SAMPLING FAILS

Conséquences désastreuses (cf. images suivantes)



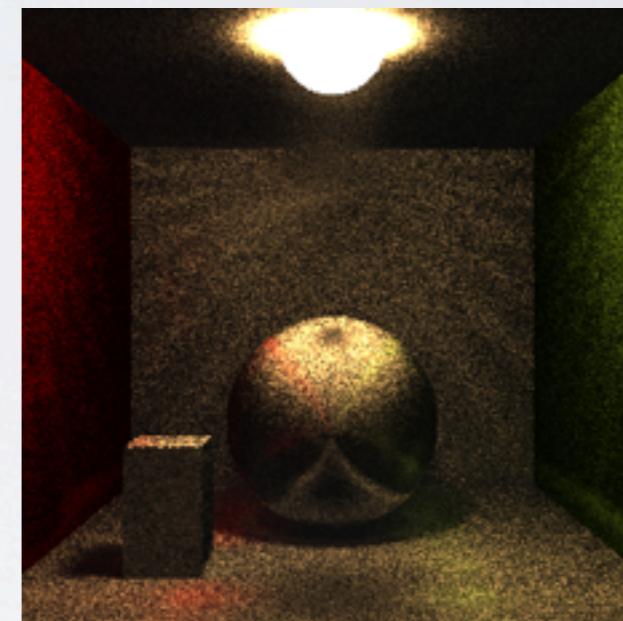
SAMPLING FAILS

X-Point star patterns
(hemisphere mapping coefficient: $e = 1.0$)



ambient occlusion

Regular @4spp × 83sets



path tracing

BRDF: jittered @900spp × 1777sets

AreaLight: jittered @100spp × 1777sets

MATERIAUX

"Sometimes you don't choose the material; the material chooses you." - JAMES D. HOUSTON

MATERIAUX UNIFORMES

- matte
- phong
- glossy reflective
- reflective
- transparent

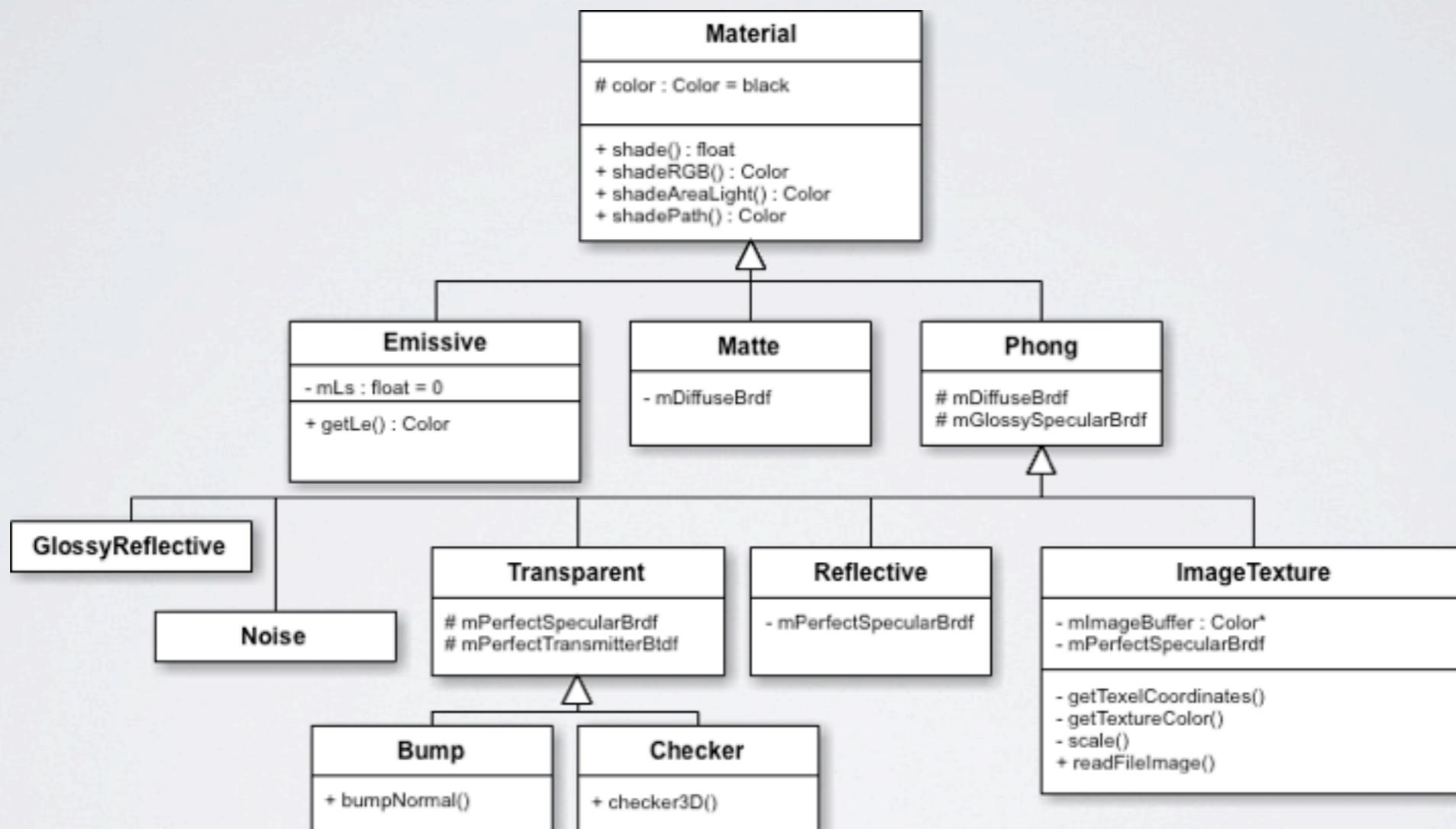
MATERIAUX TEXTURES

- procédurales
- bruits
- images
- bump-mapping

MATERIAUX EMISSIFS

- emissive

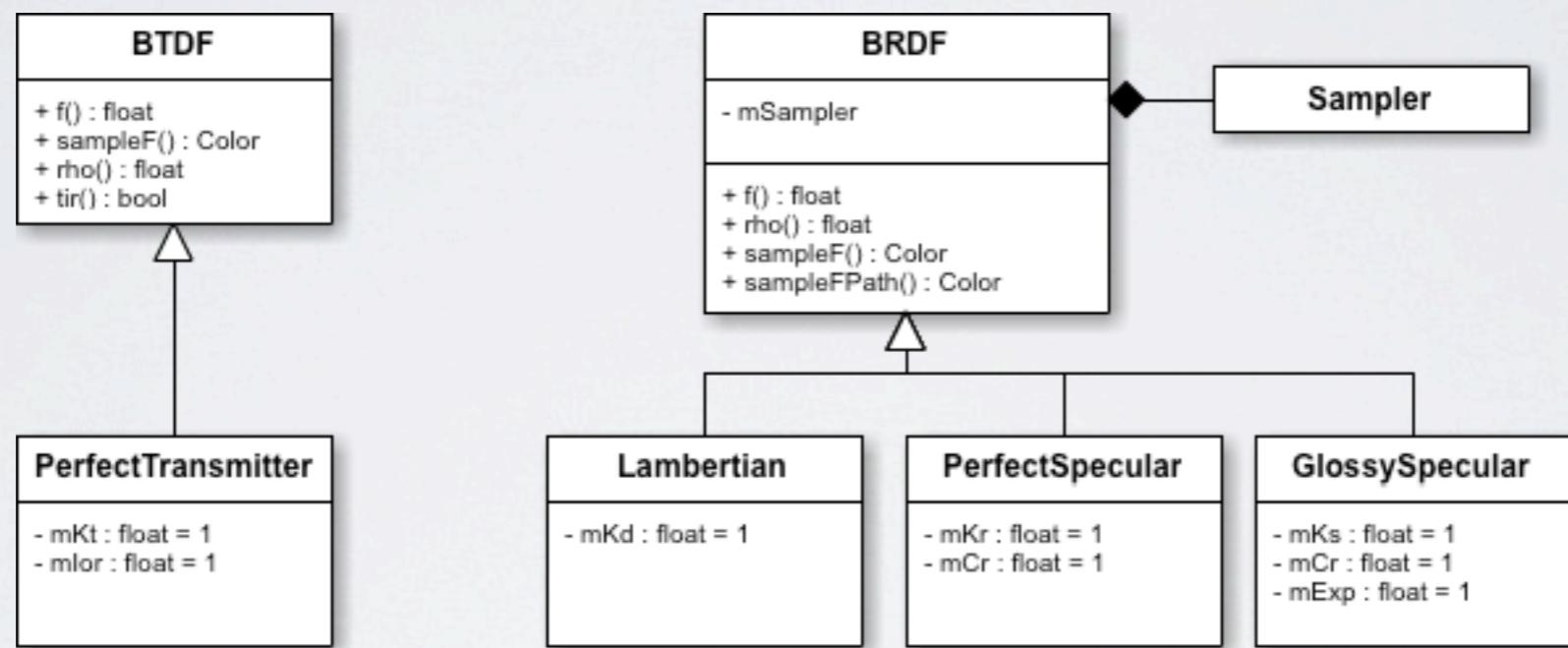
MATERIAUX



MATERIAUX

BRDF - BTDF

BRDF-BTDF



MATERIAUX

materiaux

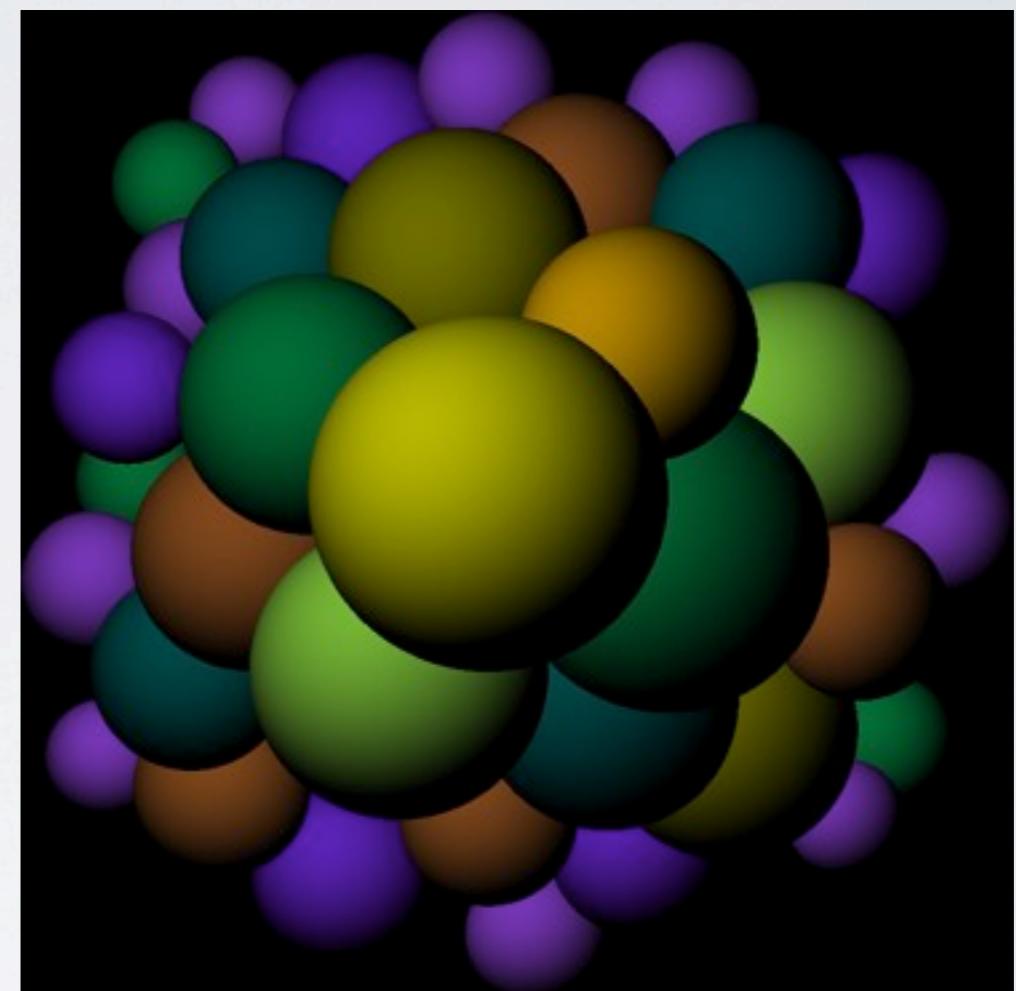
MATERIAUX

matte

Réflexion diffuse

BRDF Lambertian:

- Réflectance de Lambert
- Isotropique
- Loi du cosinus de Lambert



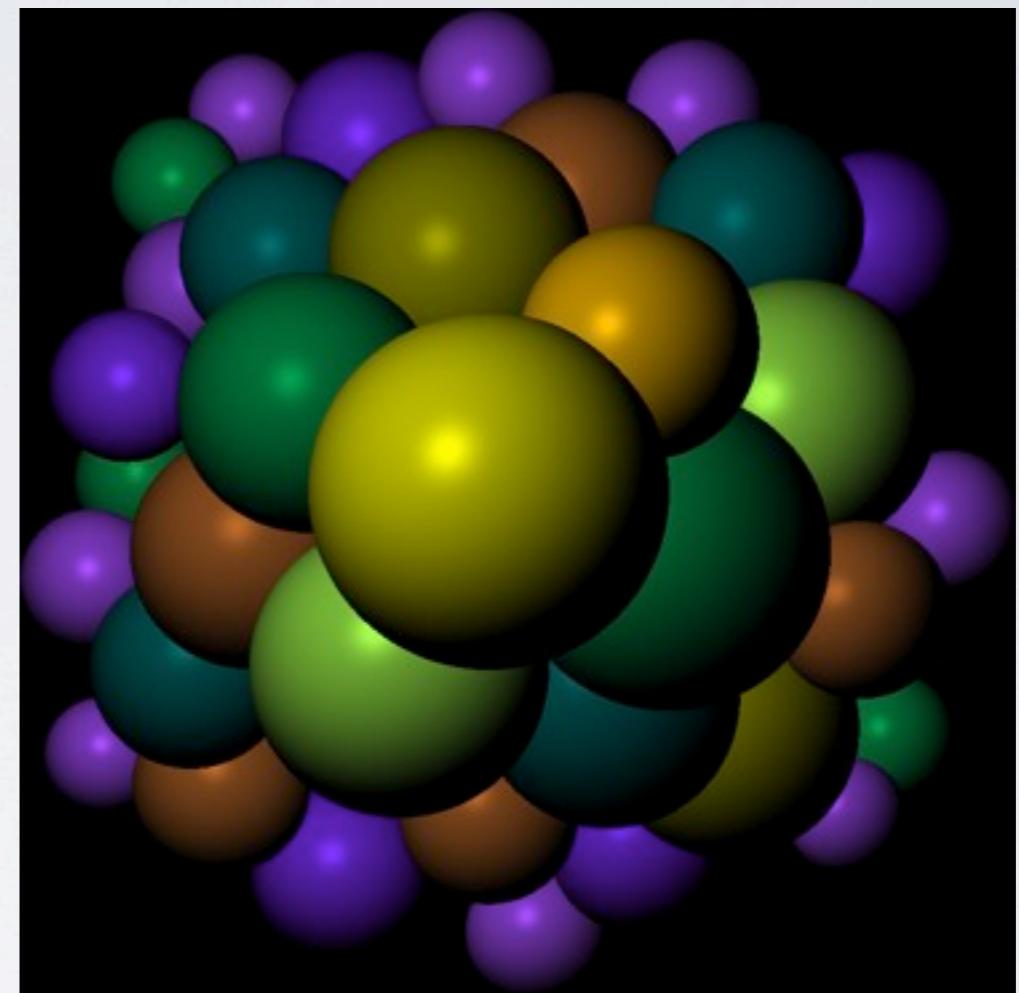
MATERIAUX

phong

PHONG

specular highlights

model de Phong

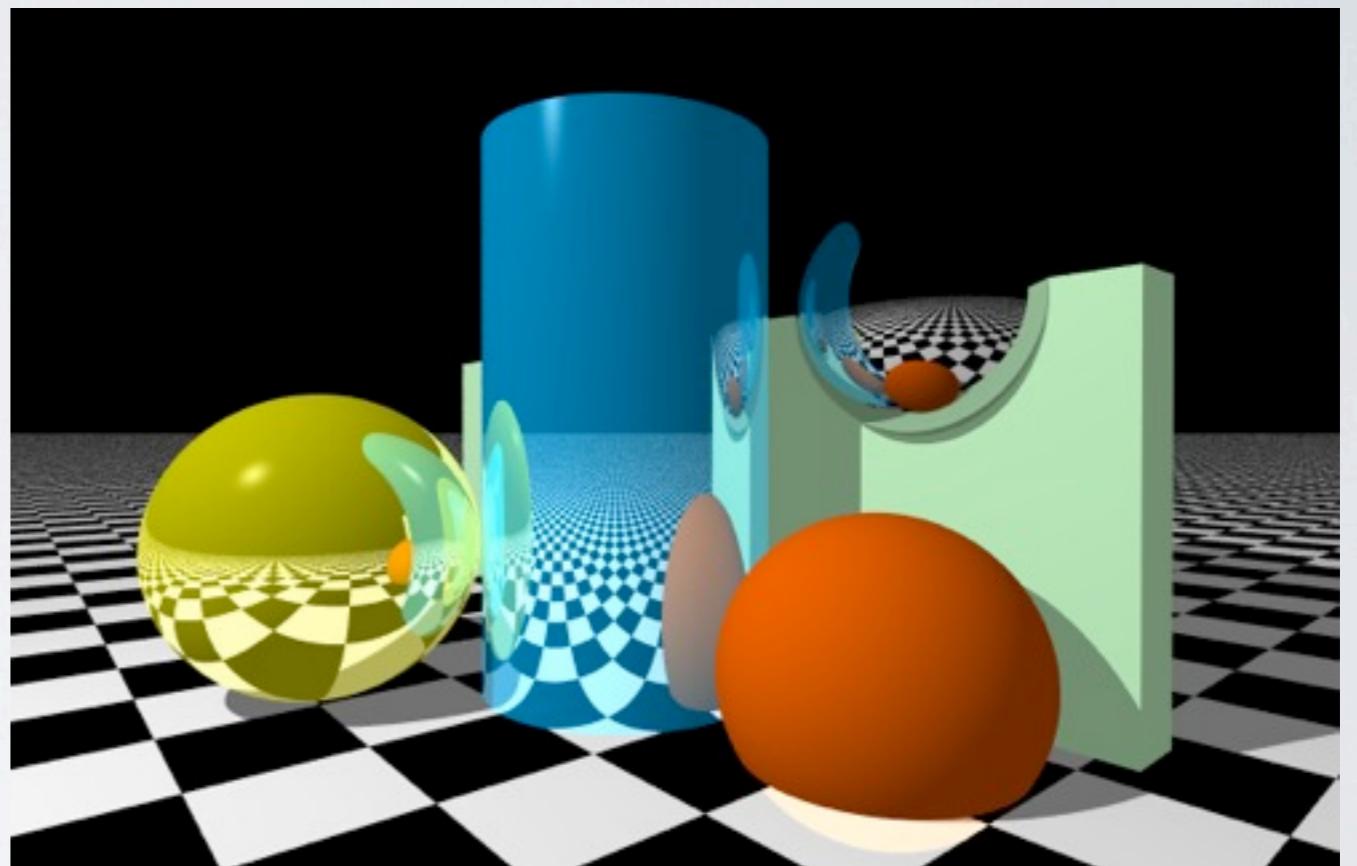


MATERIAUX

reflective

REFLECTIVE

specular reflection
(réflexion parfaite)

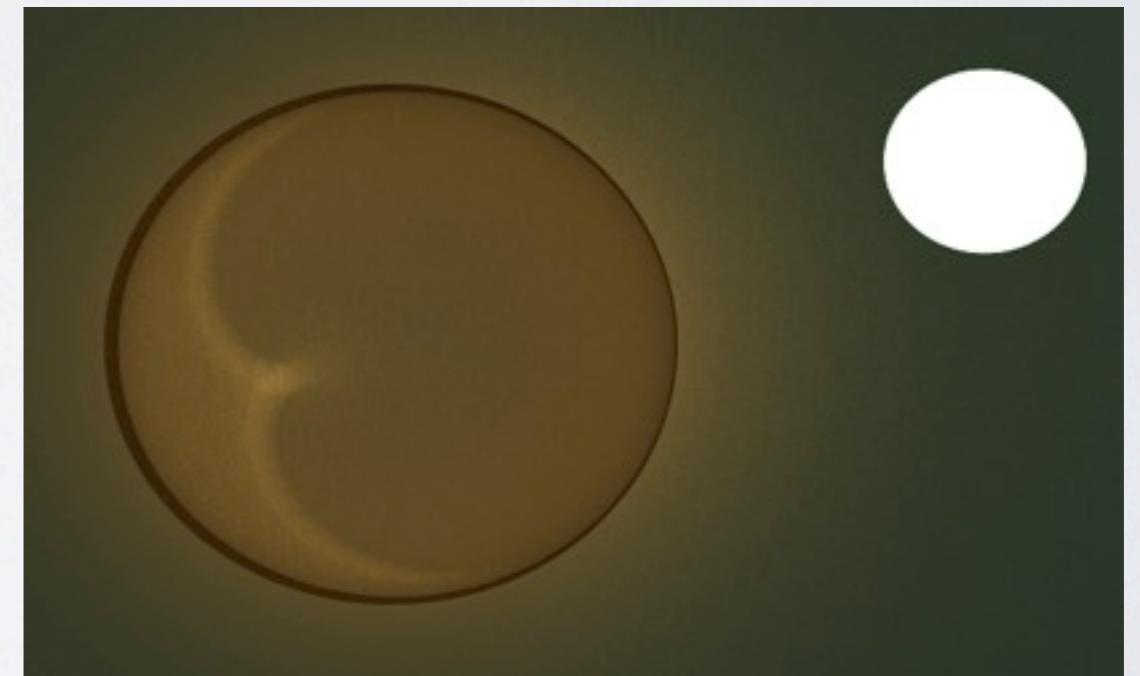


REFLECTIVE CAUSTIQUES



Photographie
d'une néphroïde

Néphroïde rendue
en path tracing

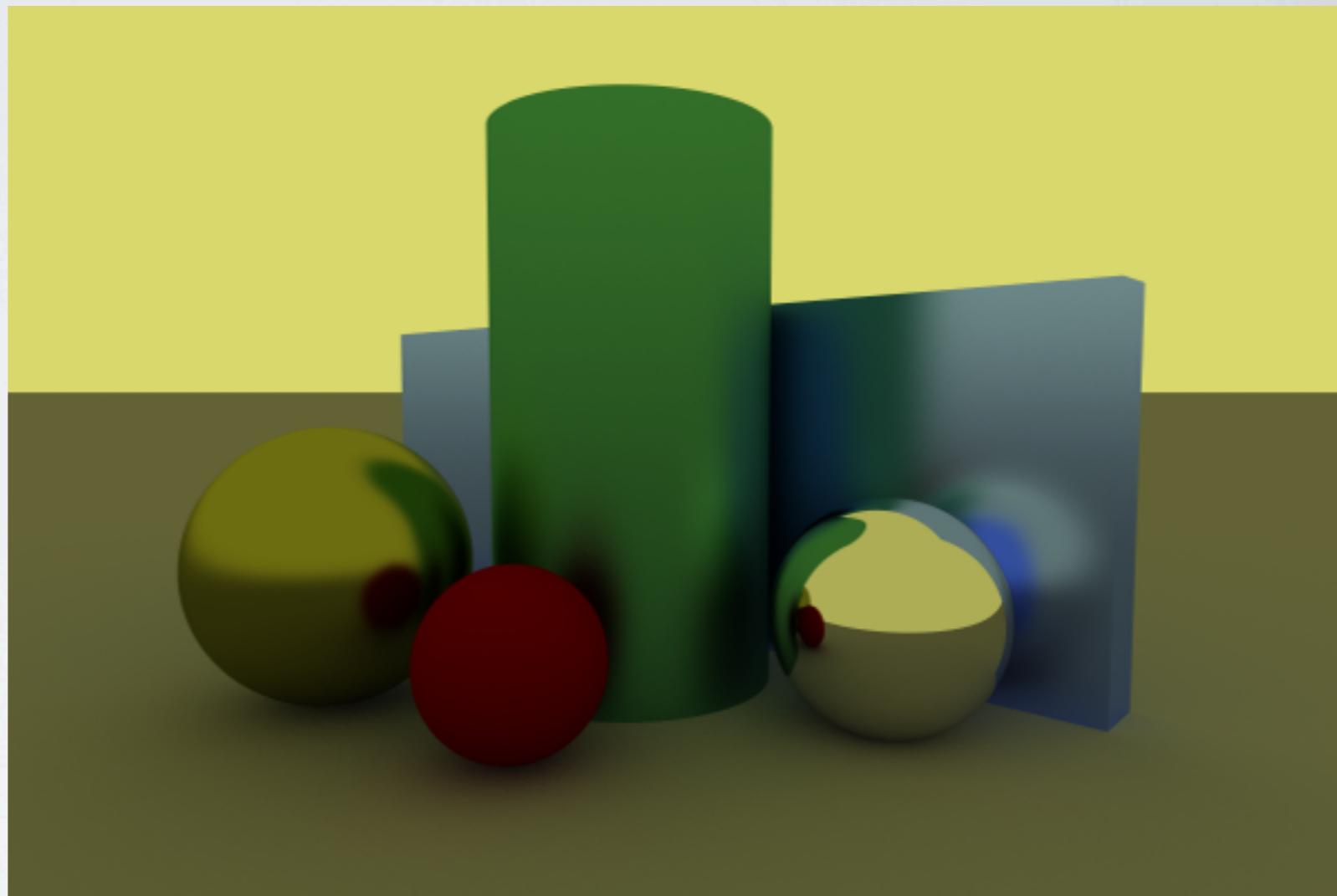


MATERIAUX

glossy reflective

GLOSSY REFLECTIVE

l'effet glossy est reproduit en appliquant des samples mappés sur hemisphere a la direction de réflexion



MATERIAUX

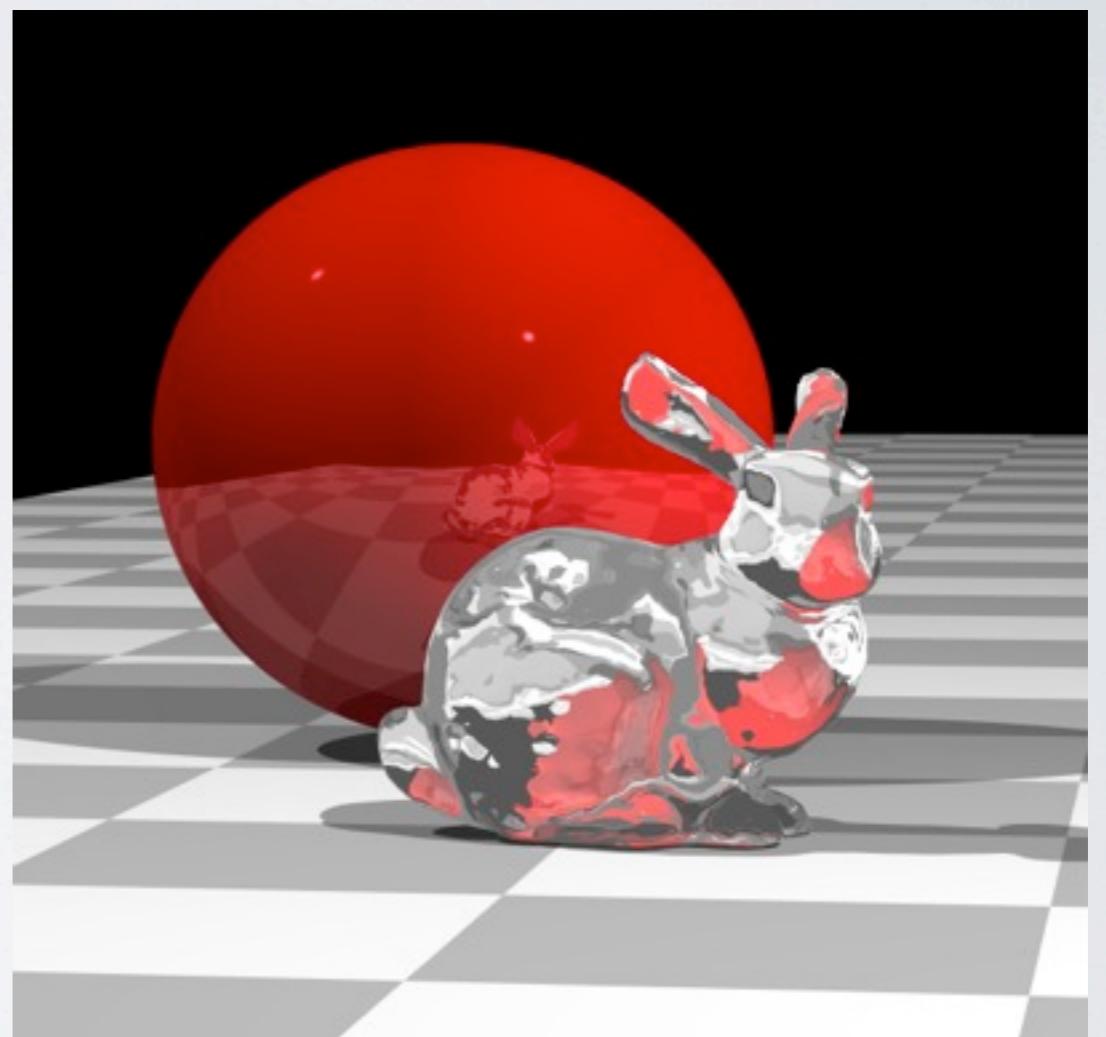
transparent

TRANSPARENT

transparence physiquement correcte

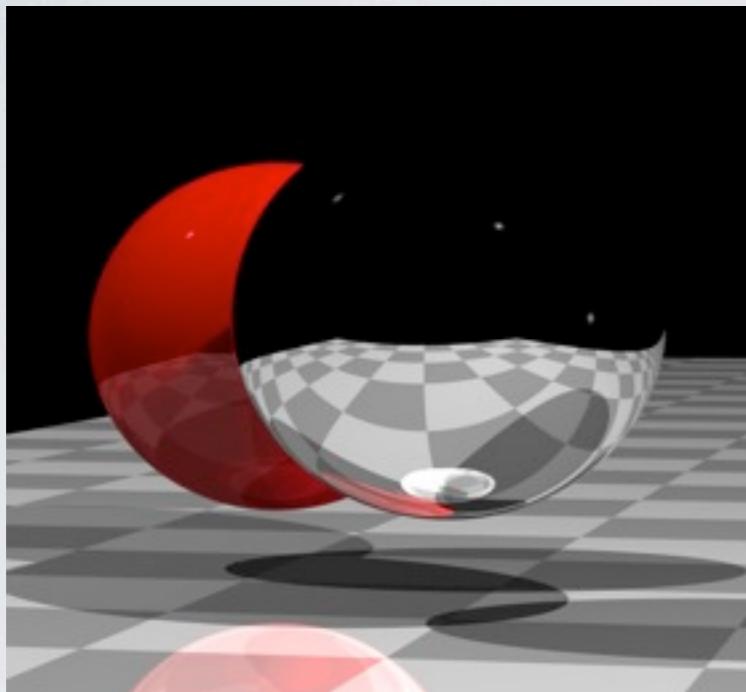
dépend de :

- coefficient de réfraction λ
- coefficient de transparence
- coefficient de reflexion

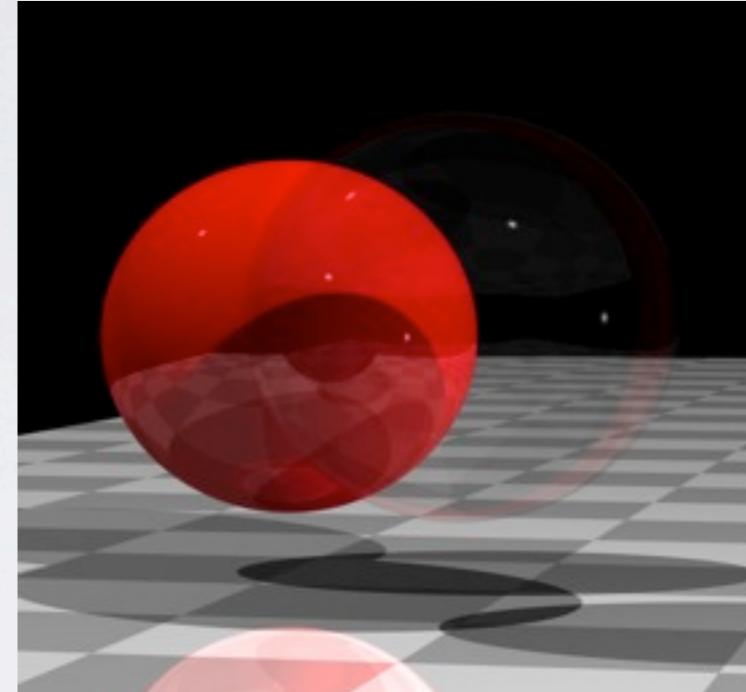


TRANSPARENT

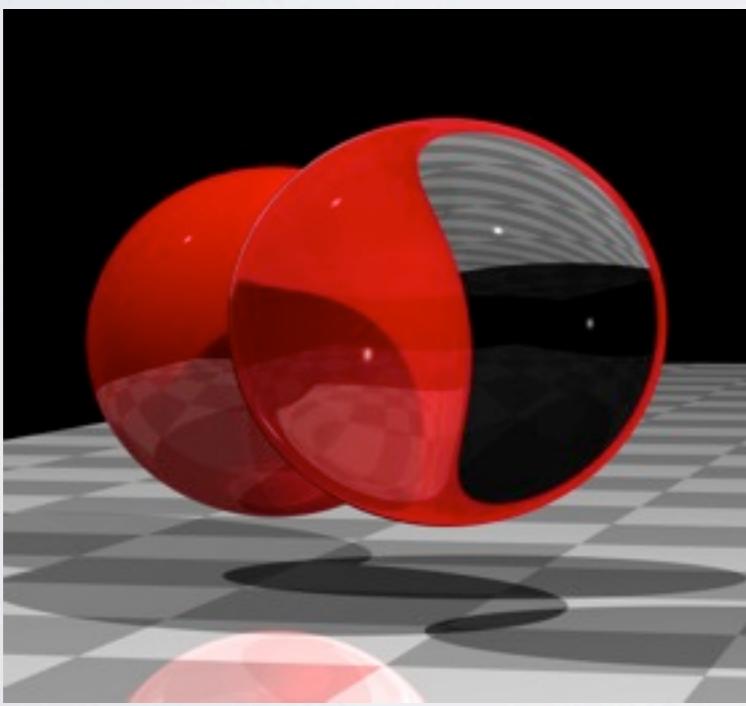
$\lambda = 0$



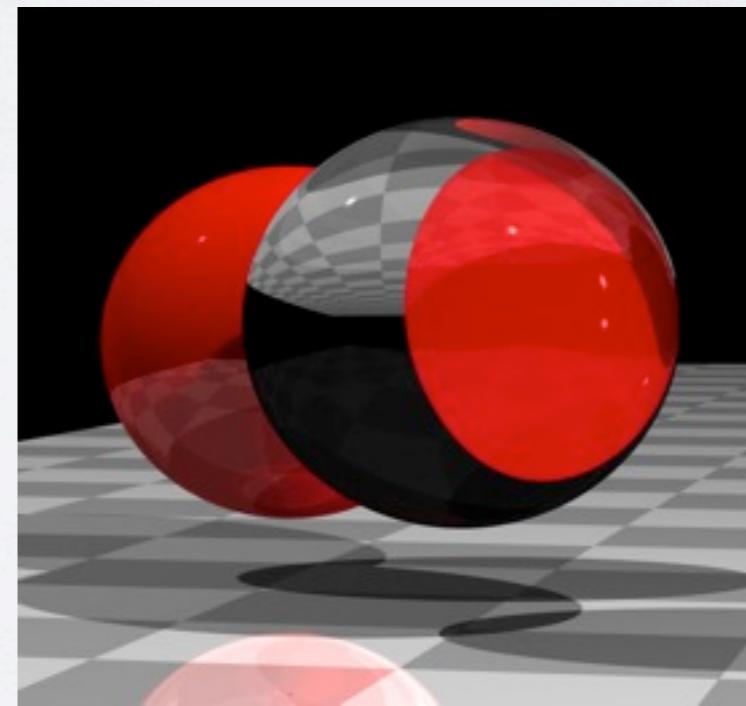
$\lambda = 1$



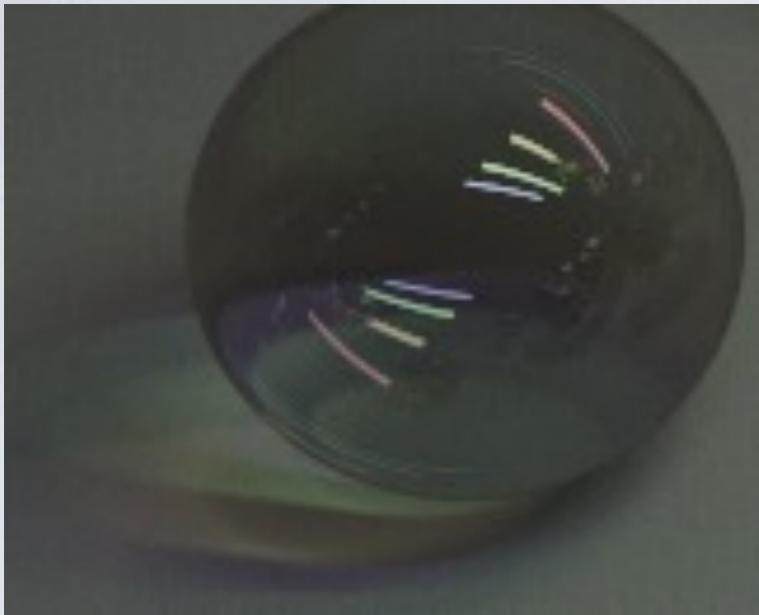
$\lambda = 1.1$



$\lambda = 1.5$



TRANSPARENT CAUSTIQUES



Caustiques rendues en path tracing

$\text{ior} = 1.10$

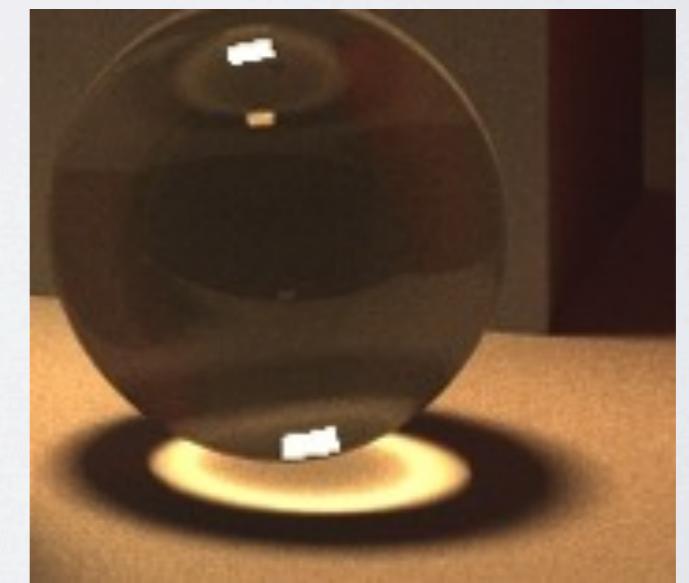
5 area lights rectangulaires



Caustique rendue en path tracing

$\text{ior} = 1.14$

1 area light rectangulaire

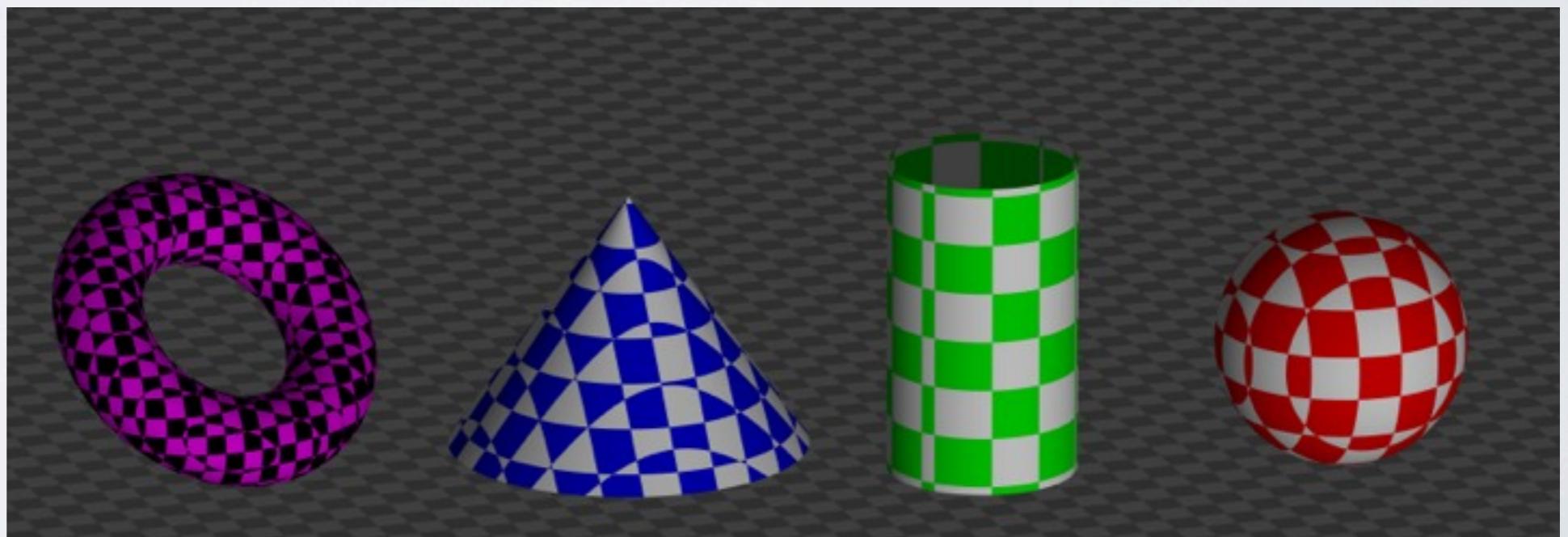


MATERIAUX

procedural

PROCEDURAL

la couleur est modifiée en
fonction des coordonnées des
points d'intersection

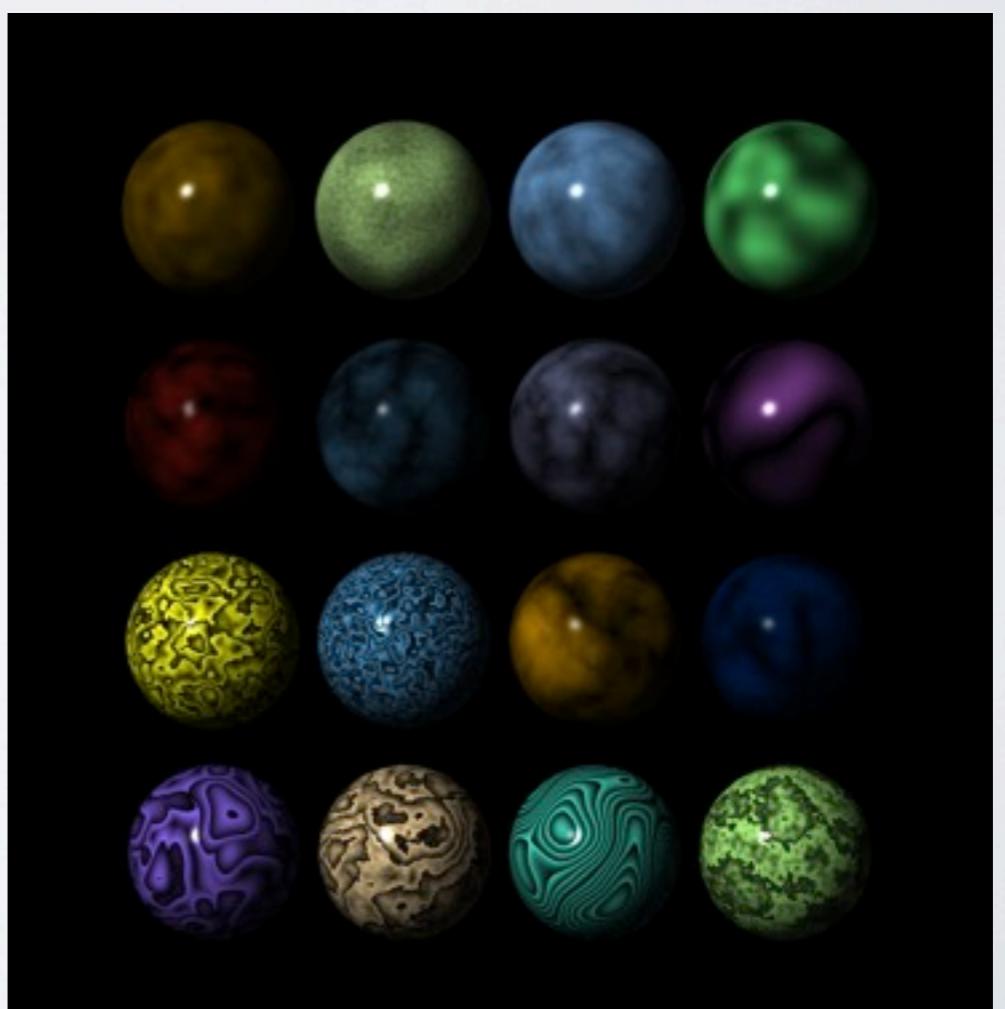
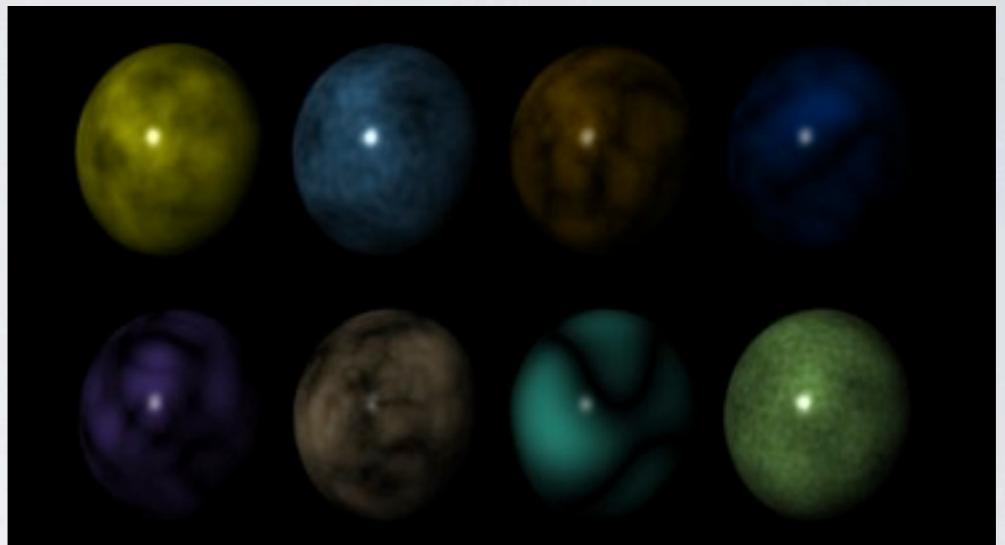


MATERIAUX

bruit

BRUIT

la couleur est modifiée en fonction d'un bruit cohérent obtenu à partir des coordonnées des points d'intersection

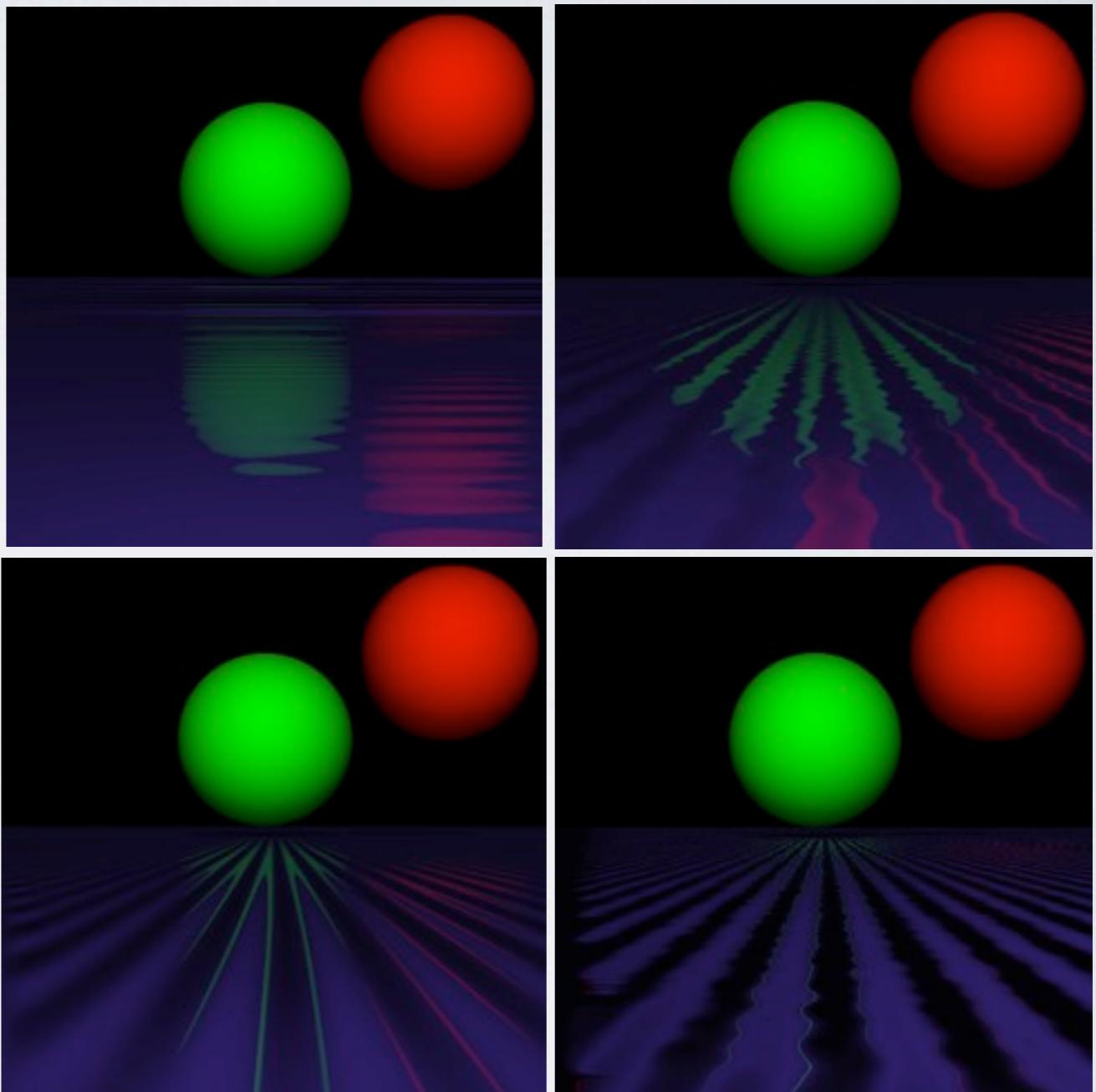


MATERIAUX

bump mapping

BUMP MAPPING

la modification de la normale sur un ou plusieurs axes entraîne une modification de la luminosité et un effet de relief

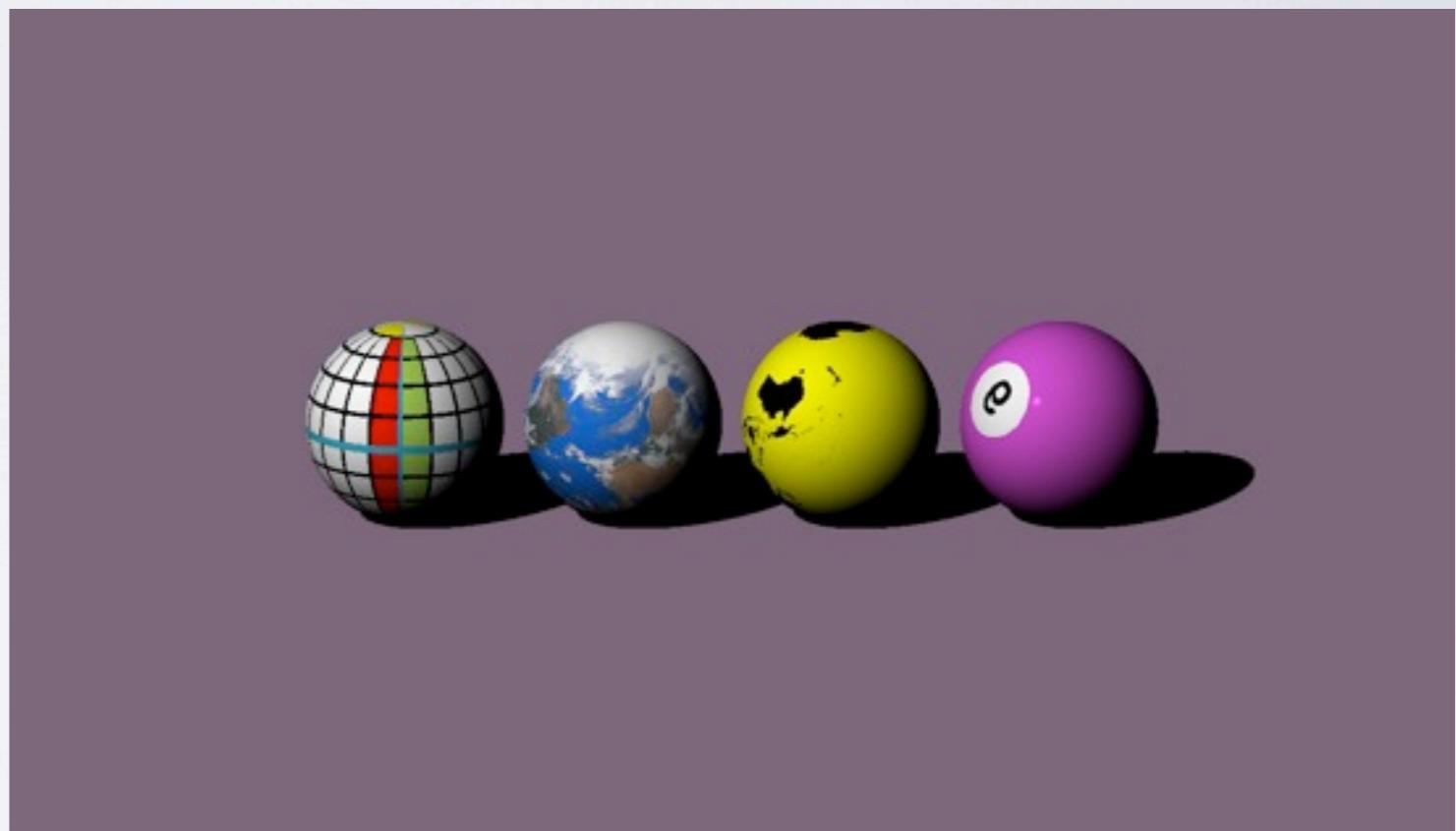


MATERIAUX

image mapping

IMAGE MAPPING

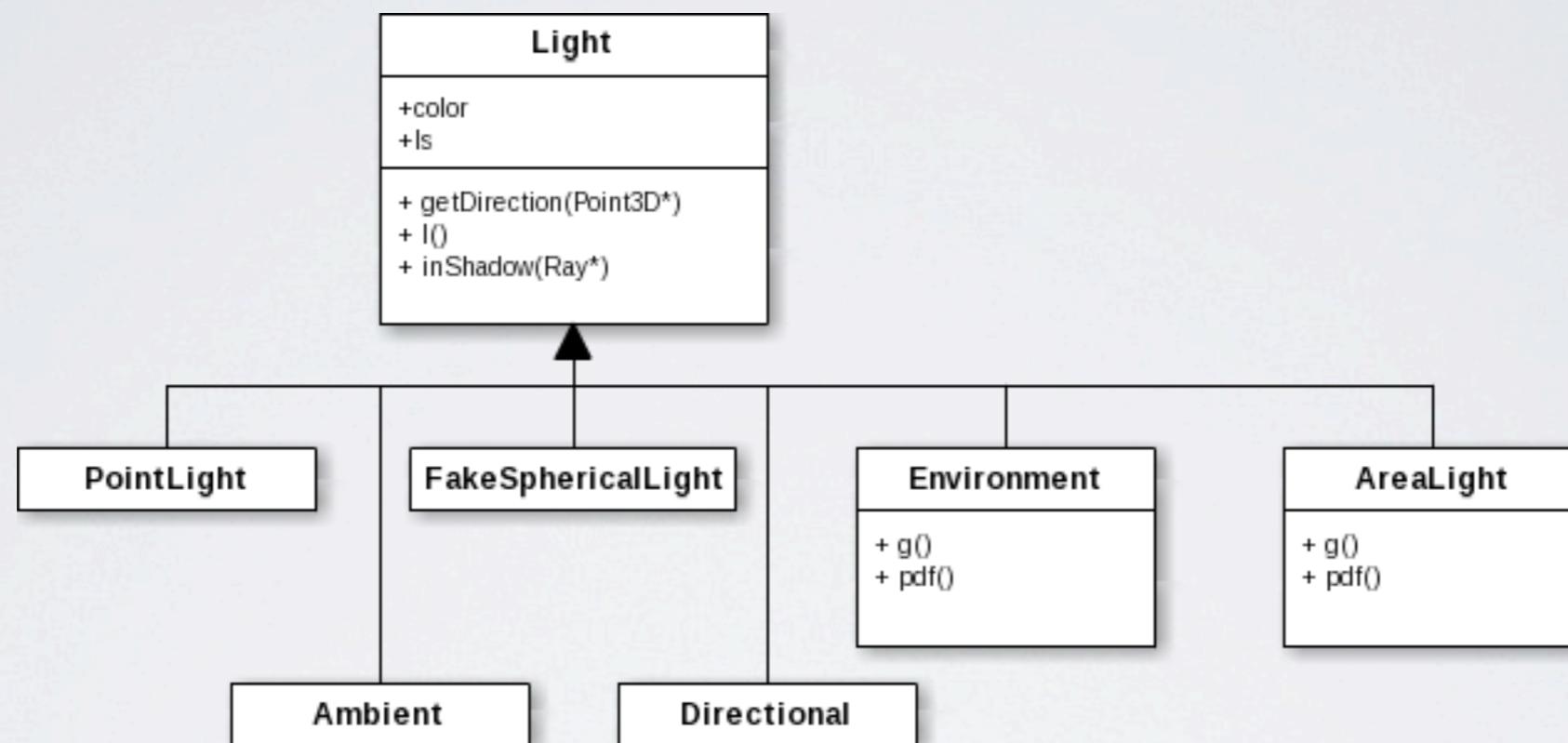
À partir des coordonnées 3D(x, y, z) on obtient les coordonnées sphérique (θ, ϕ) qui nous servent ensuite à obtenir des coordonnées 2D (u, v) pour accéder à la couleur de la texture.



LUMIERES

"Every moment of light and dark is a miracle." - WALT WHITMAN

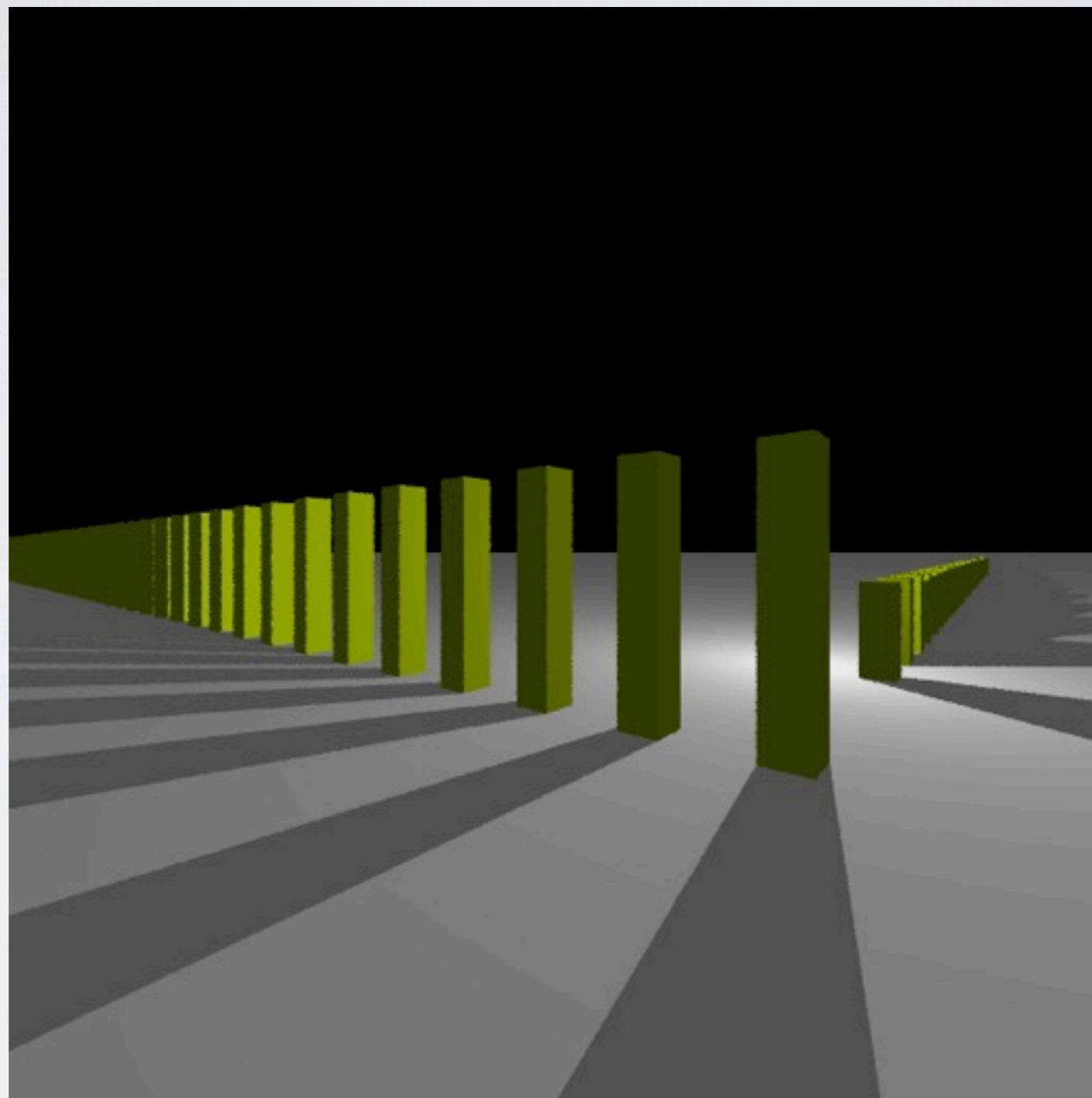
LUMIERES



LUMIERES

point

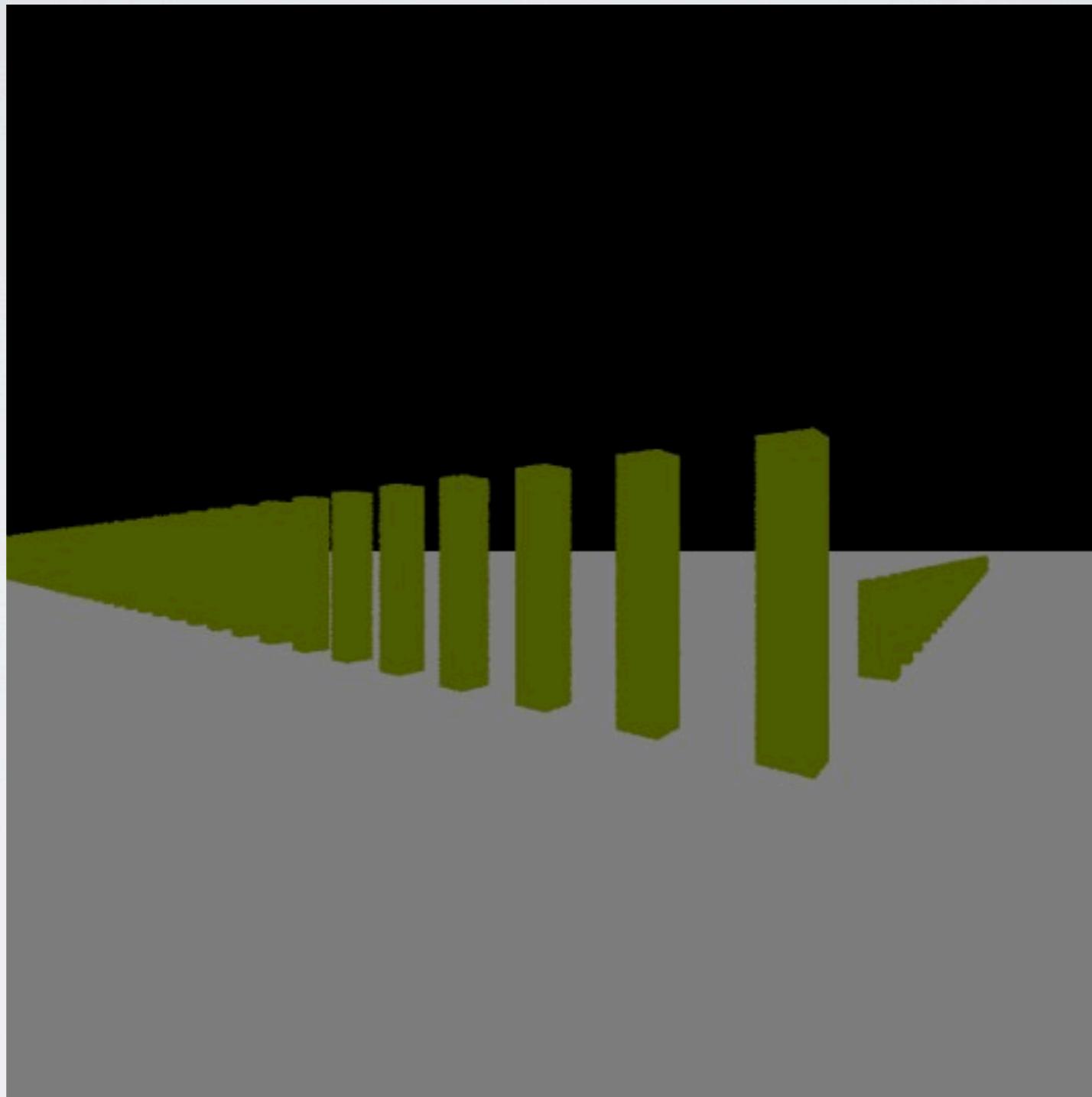
POINT



LUMIERES

ambient

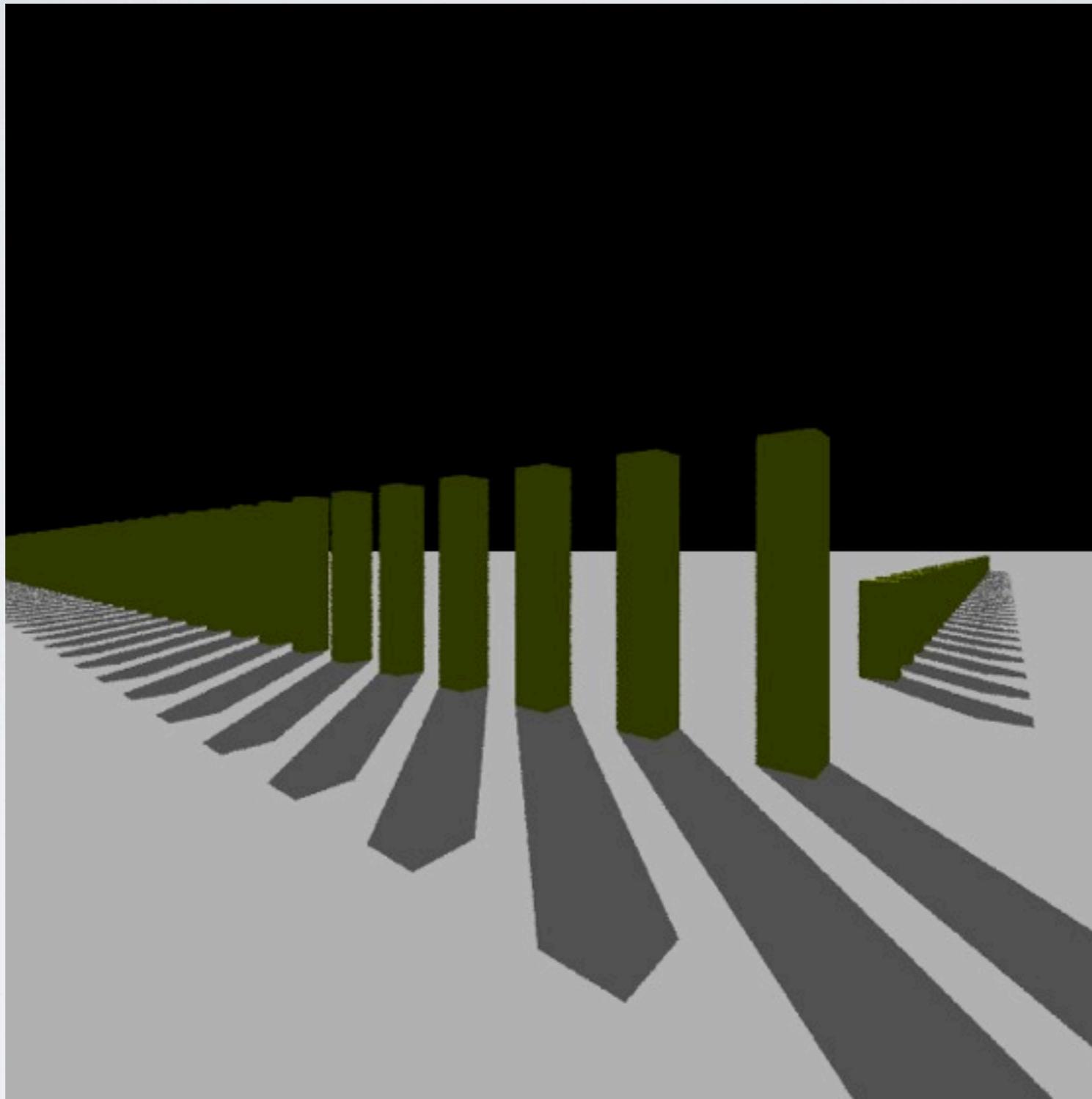
AMBIENT



LUMIERES

directional

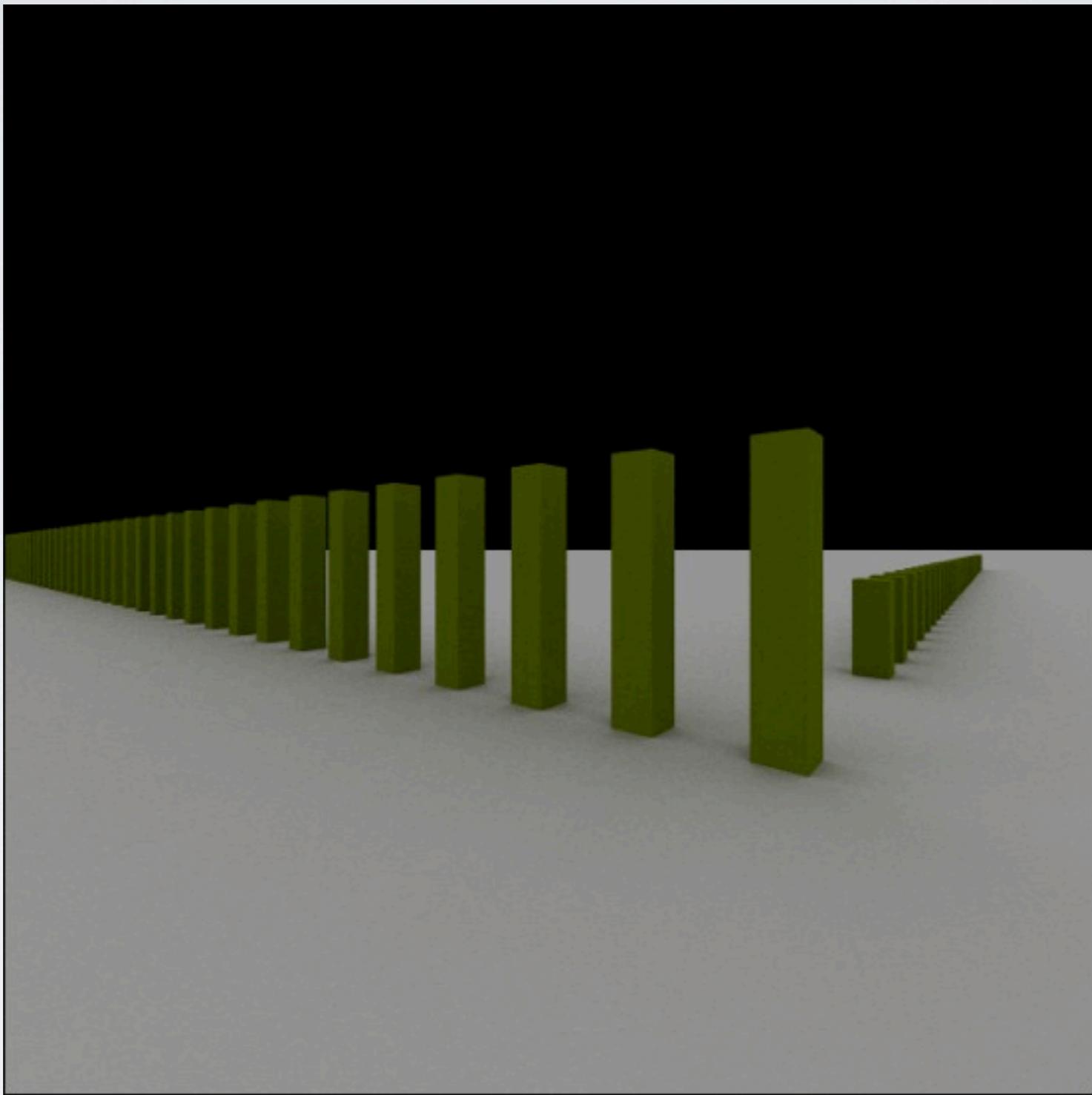
DIRECTIONAL



LUMIERES

environment

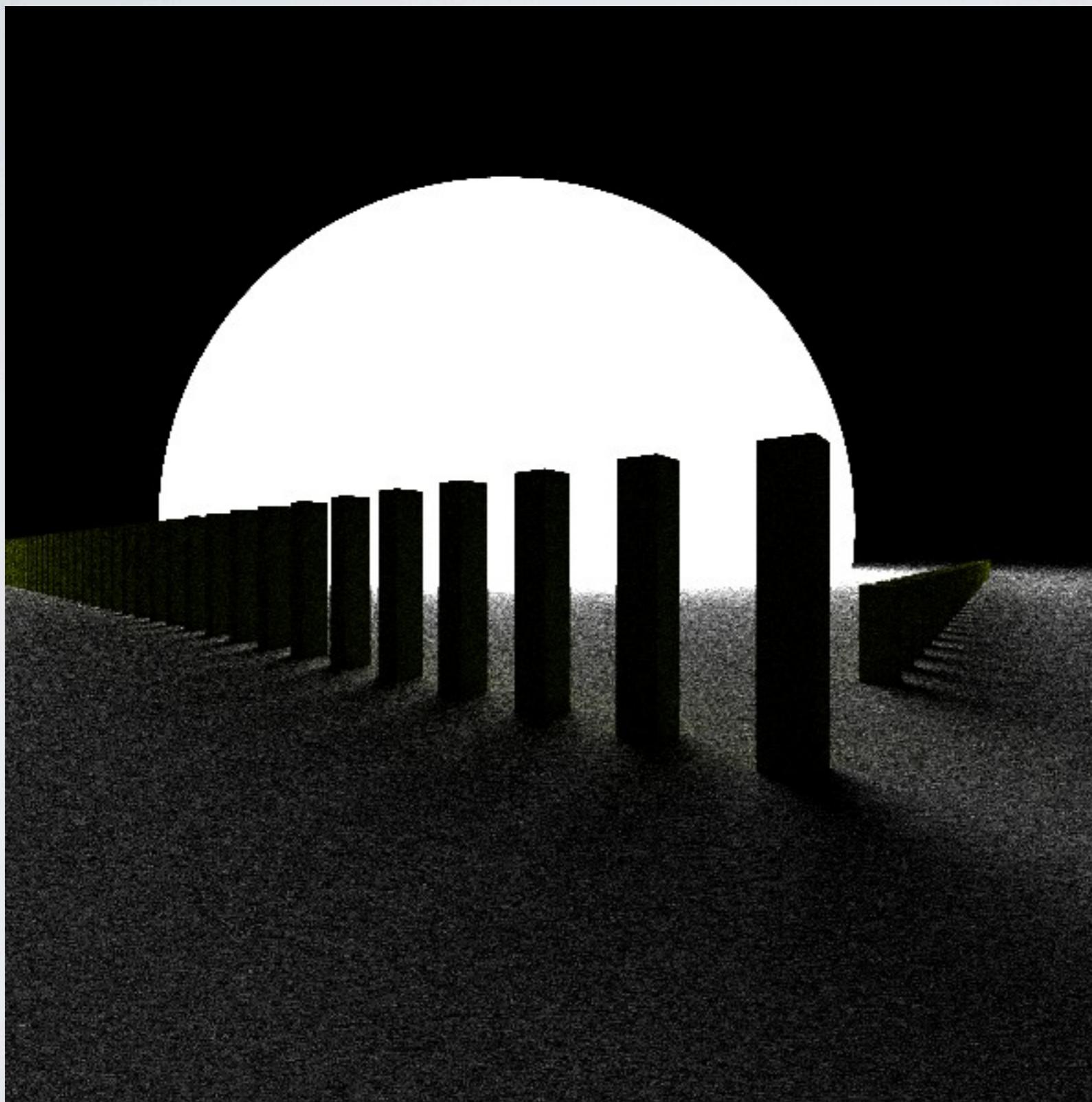
ENVIRONMENT



LUMIERES

area

AREA

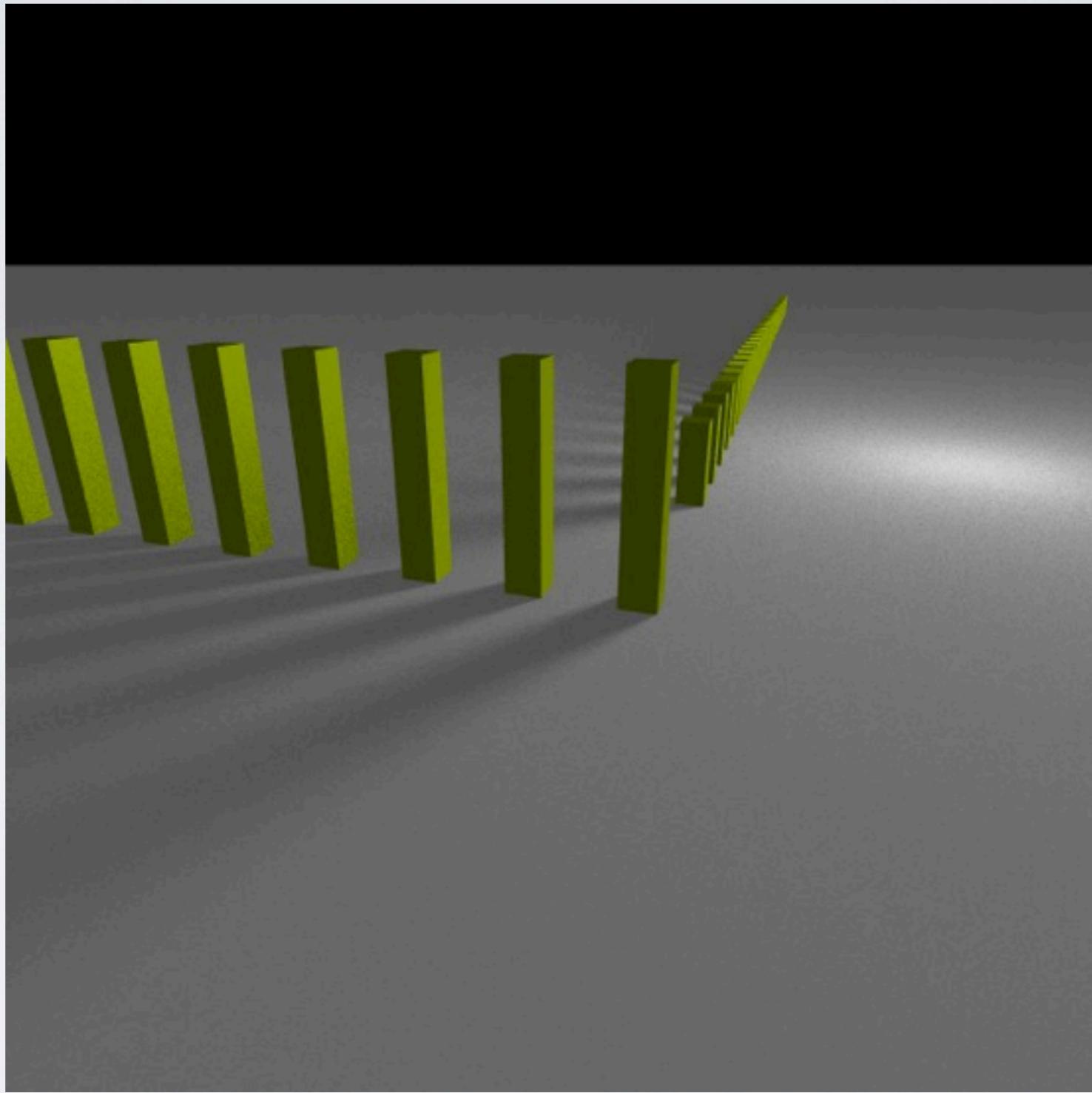


Path tracing @ 100x257
AreaLight @ 400x257

LUMIERES

fake spherical

FAKE SPHERICAL



RENDERPASS

"Always render more and better images than is expected of you, no matter what your task may be." - OG MANDINO

RENDERPASS

tâche spécifique appliquée à l'ensemble des pixels de l'image

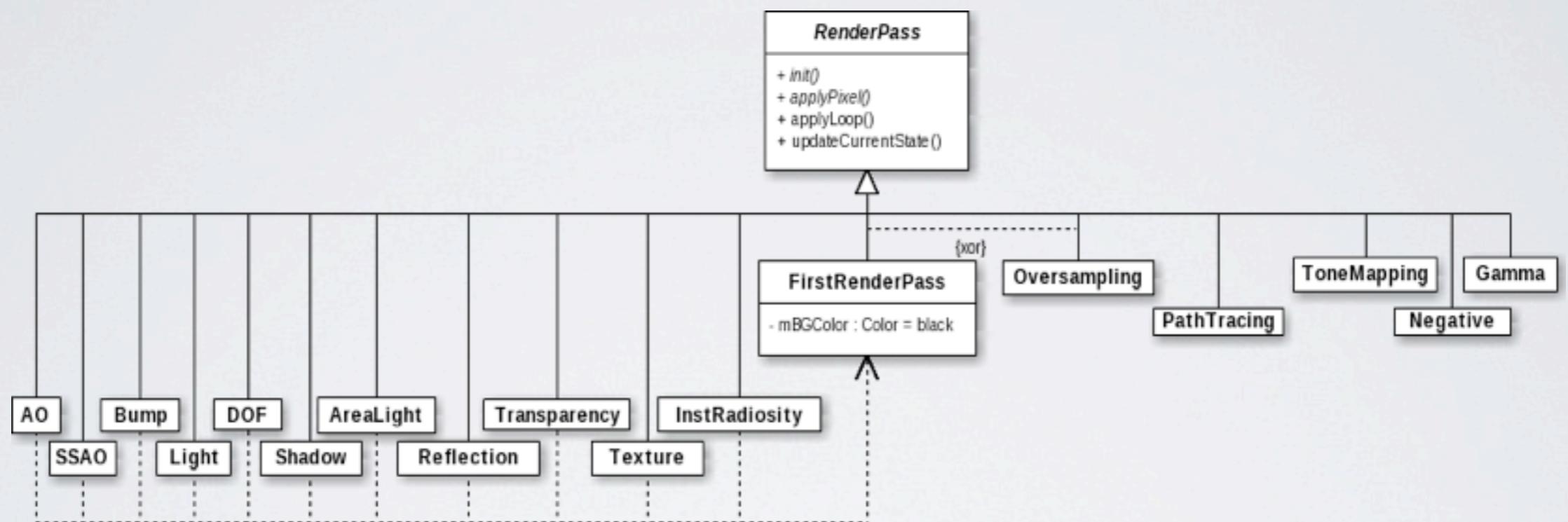
associée à un (voire plusieurs) buffer(s)

parfois associée à un matériel (texture, transparence)

permet:

- d'ajouter très facilement de nouvelles techniques
- d'activer/désactiver très facilement les techniques utilisées
- de pouvoir voir les différentes étapes du rendu

RENDERPASS



RENDERPASS

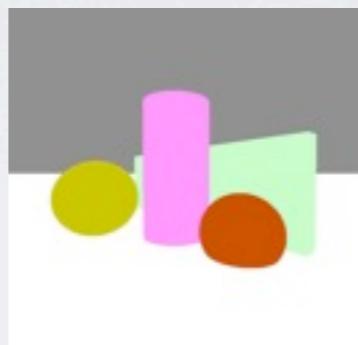
buffers

- primordiaux
- associent une valeur (vecteur, rayon, couleur, radiance, ..) à une coordonnée dans l'image
- permettent d'implémenter des techniques en screen space (SSAO, ..)

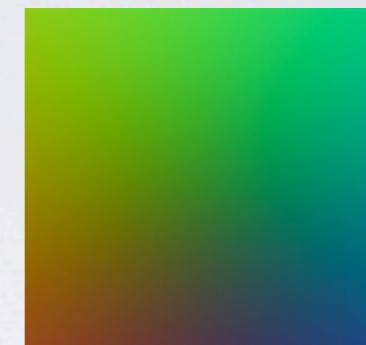
ensemble de buffers fondamentaux



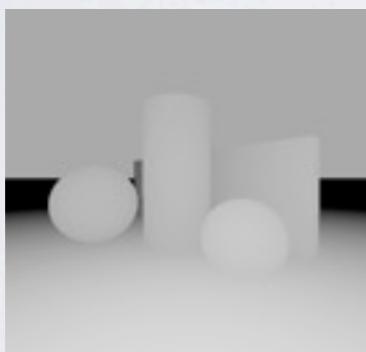
normalbuffer



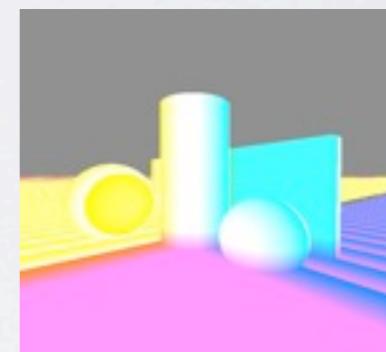
colorbuffer
materialbuffer



raybuffer

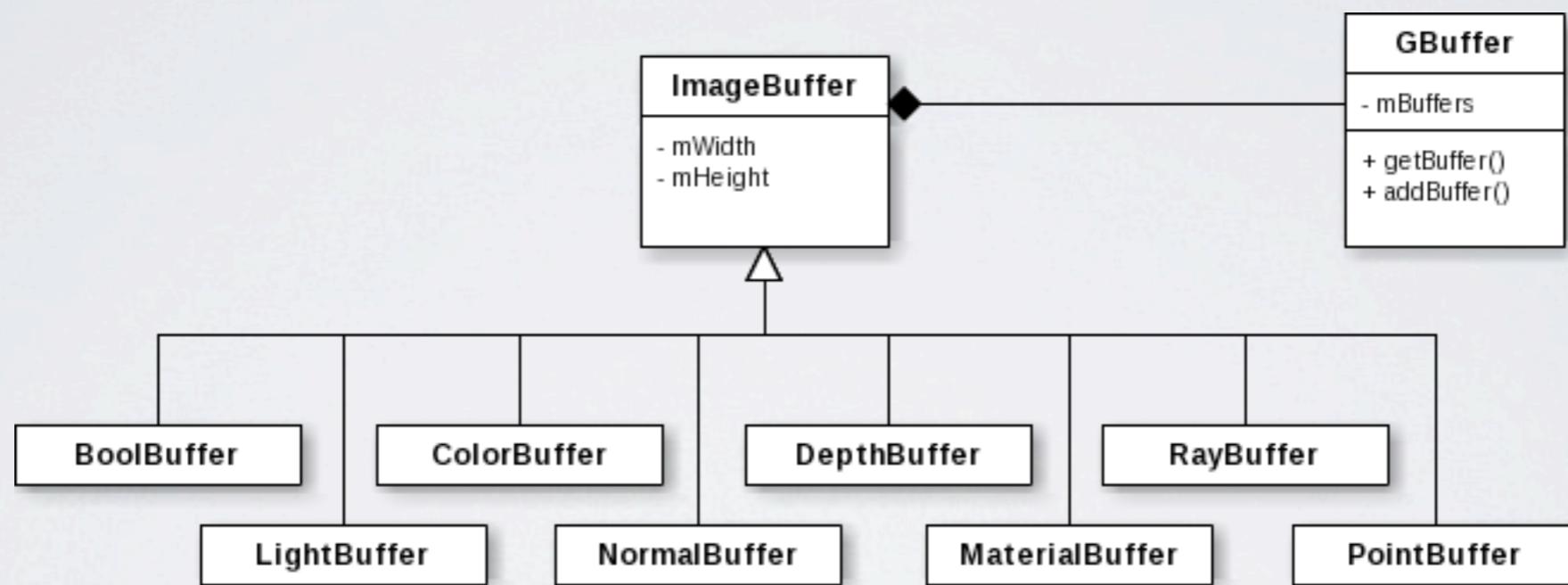


depthbuffer (zbuffer)



hitpointbuffer

BUFFERS



RENDERPASS

renderpasses

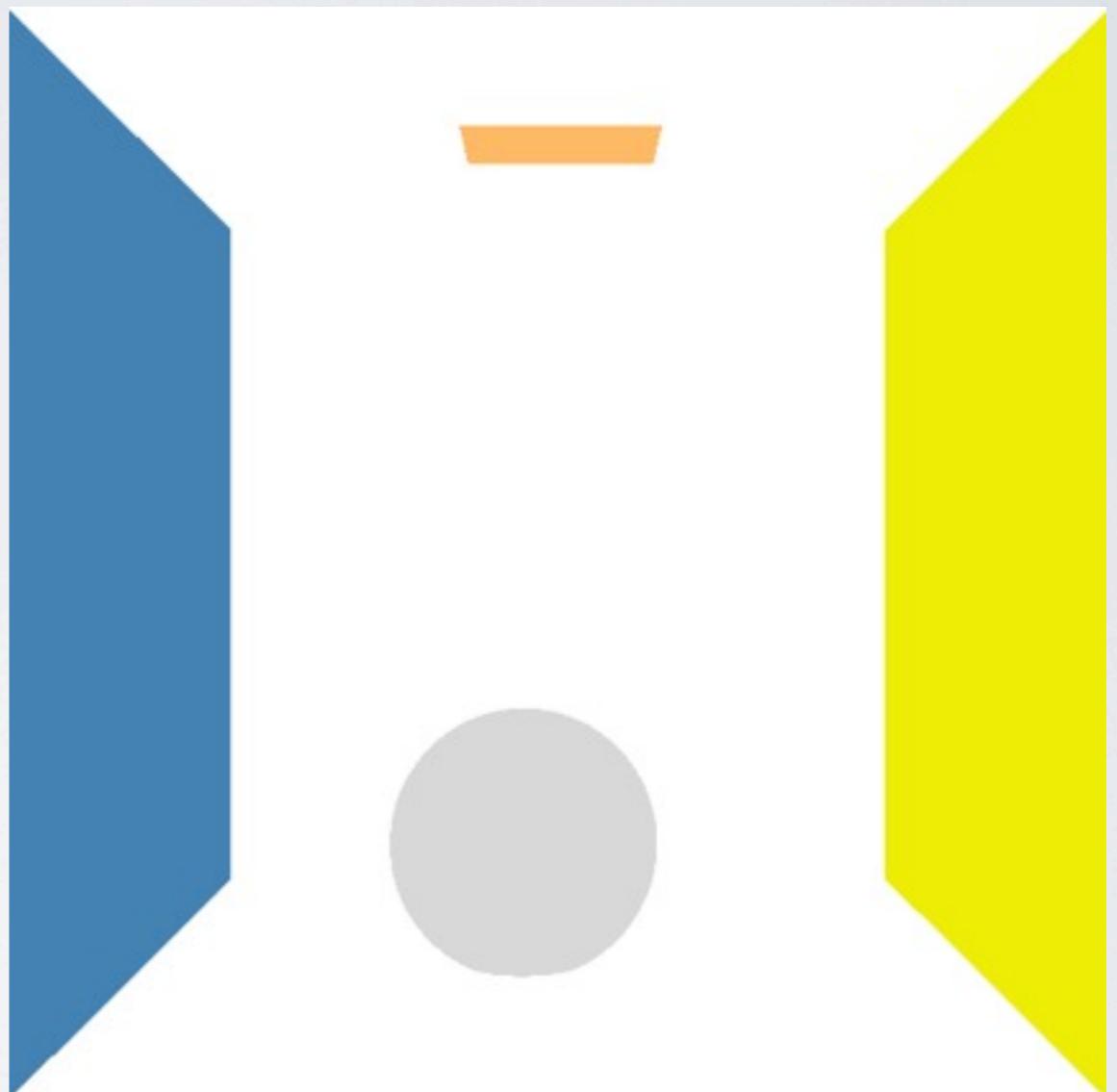
RENDERPASS

first renderpass

FIRST RENDERPASS

remplit tout les buffers fondamentaux (gbuffer)

obligatoire pour la majorité des autres renderpasses



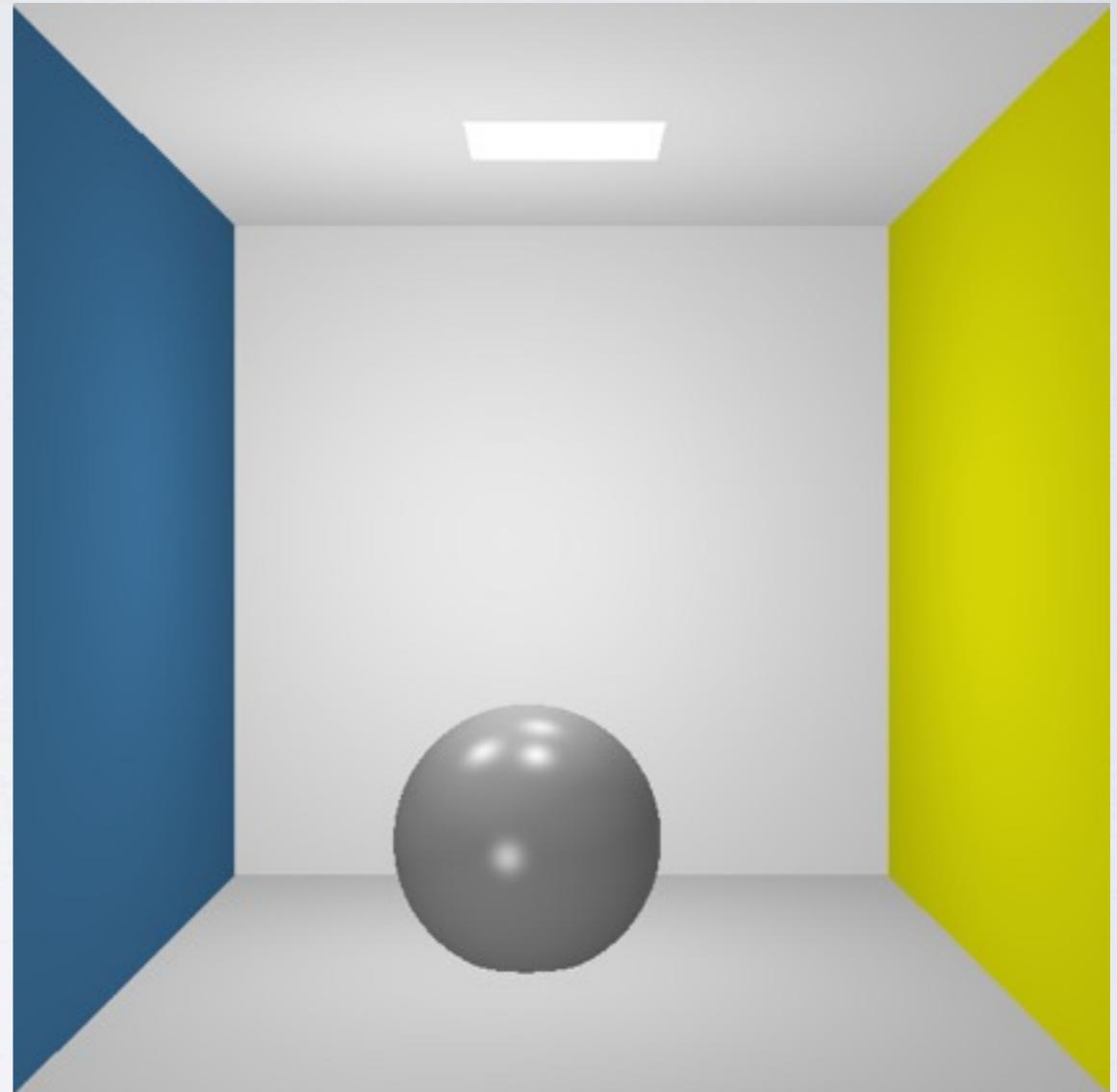
RENDERPASS

light

LIGHT

remplit un lightbuffer pour chaque lumière

obligatoire pour certaines renderpass (ombres, AO, SSAO, ..)

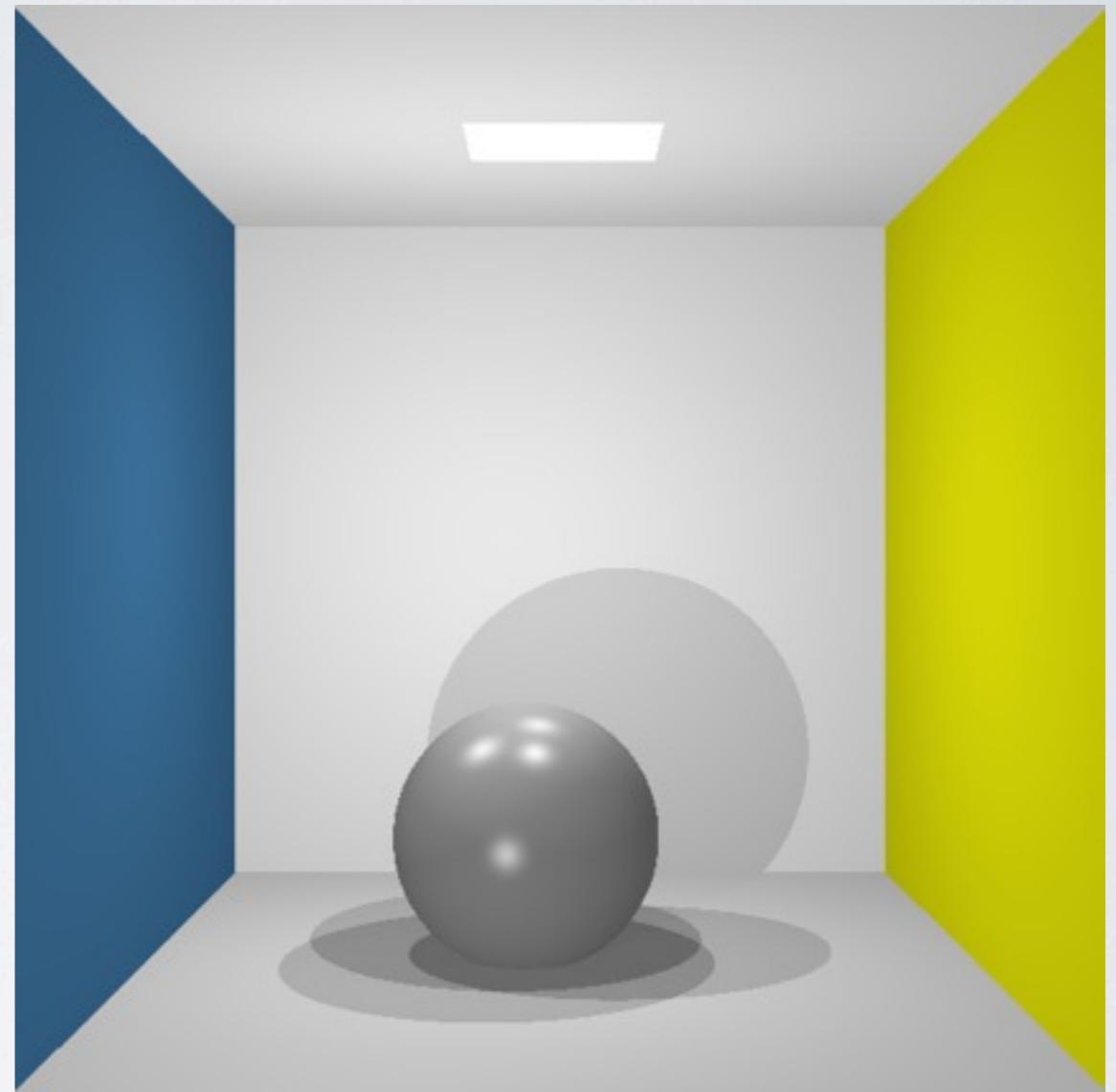


RENDERPASS

shadow

SHADOW

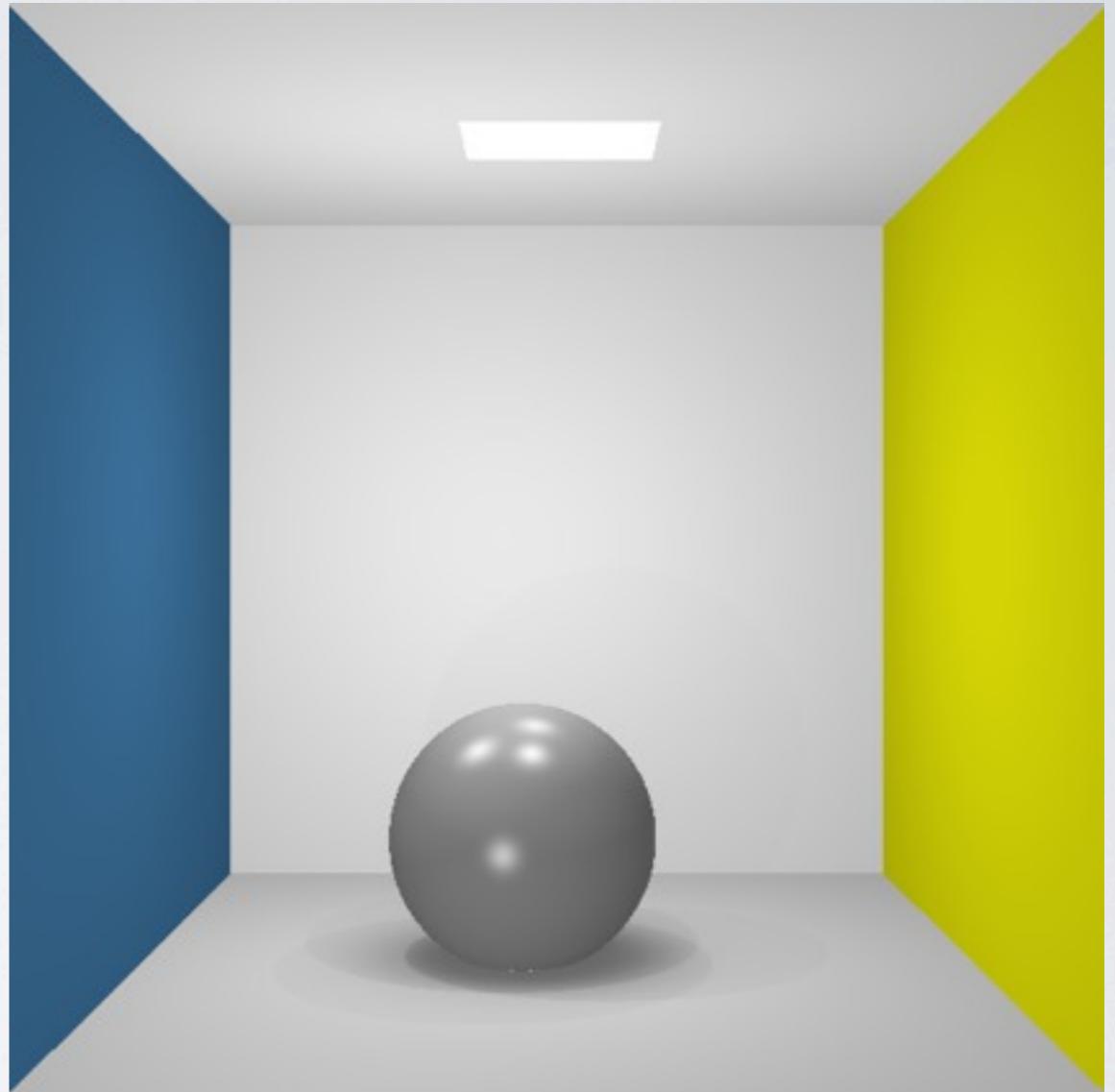
éteint certains pixels des
lightbuffers



SHADOW SOFT SHADOWING

atténue les lightbuffers

se base sur la distance du point
occlué à la lumière

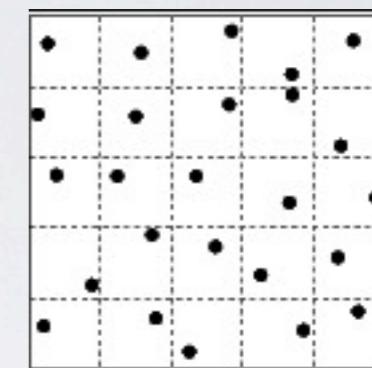
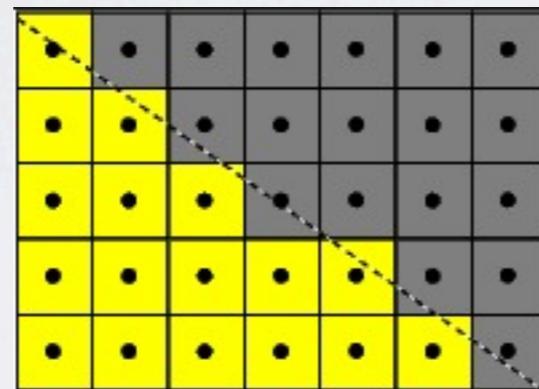
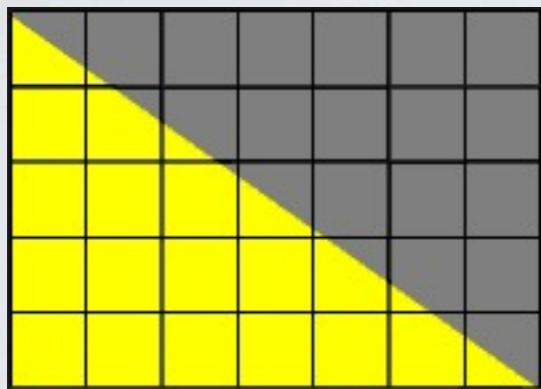


RENDERPASS

oversampling

OVERSAMPLING

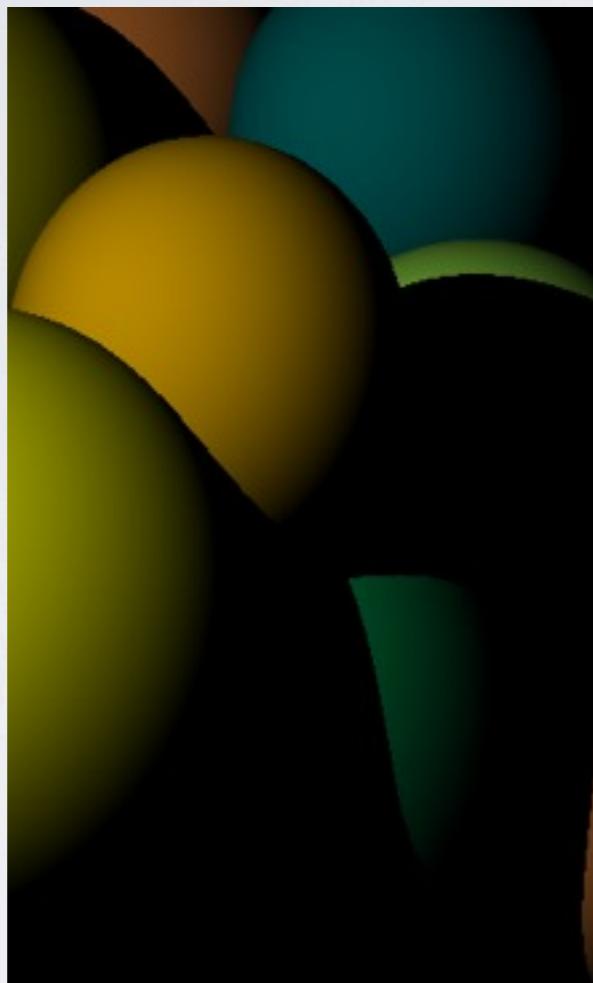
pour éviter l'*aliasing*, il faut répartir plusieurs samples (lancer plusieurs rayons) dans un même pixel



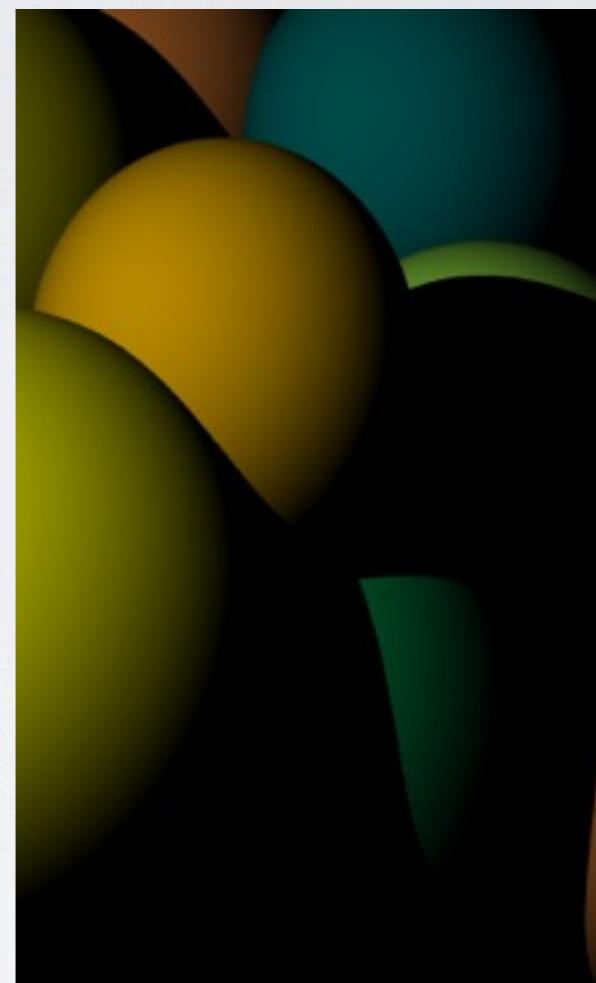
L'*aliasing* qui a lieu quand un seul sample est utilisé par pixel

25 samples dans un pixel
(25 spp / Jittered)

OVERSAMPLING



1 spp



25 spp

RENDERPASS

screen space ambient occlusion

- résultats similaires à de l'ambient occlusion, en screen-space
- technologie récente (2007 - à la pointe du rendu 3D realtime, introduite par CryTek, pour CryEngine 2+)
- rendu instantané
- vitesse constante (ne dépend pas de la géométrie de la scène)
- ultra-configurable, donc plusieurs effets possibles
- beaucoup plus adapté aux scènes d'intérieur
- permet de gagner énormément en réalisme, avec un coût négligeable en performances

technique ultra-configurable:

nombre de samples (coordonnées adjacentes ou les normales, coordonnées, et profondeurs seront lues)

rayon de sampling (pixels)

falloff (intensité minimale, voir fonctions {,smooth}step de GLSL (OpenGL) et HLSL (DirectX))

intensité

coefficients multiplicateurs

taille du kernel du flou gaussien

SSAO

ambient light uniquement
rendu instantané



Sans SSAO



Avec SSAO
MultiJittered @16x83
falloff = 0.1, rad = 1px



Avec SSAO
MultiJittered @16x83
falloff = 0, rad = 2.5px

SSAO

ambient light uniquement
rendu instantané



Sans SSAO



Avec SSAO
MultiJittered @16x83
falloff = 0, rad = 1px



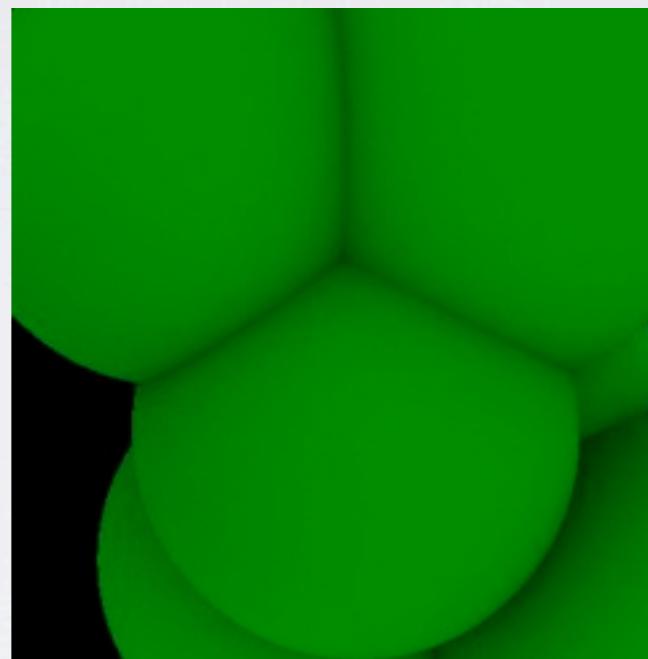
Avec SSAO
MultiJittered @16x83
falloff = 0.1, rad = 2px

RENDERPASS

ambient occlusion

AO

ambient light uniquement
quelques minutes



Multijittered @100x83

AO

ambient light uniquement
~10 minutes



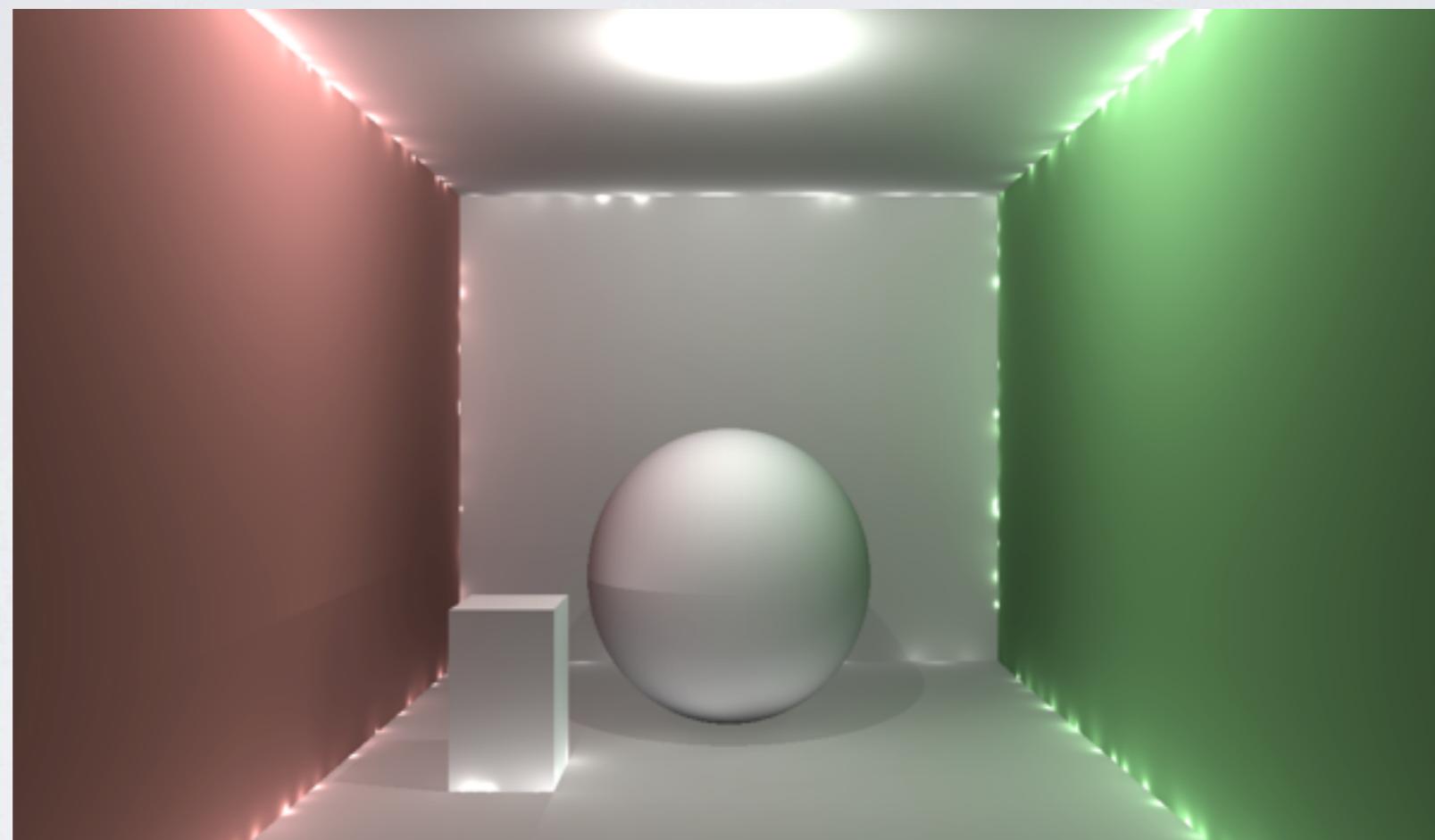
Multijittered @100x83

RENDERPASS

instant radiosity

INSTANT RADIOSITY

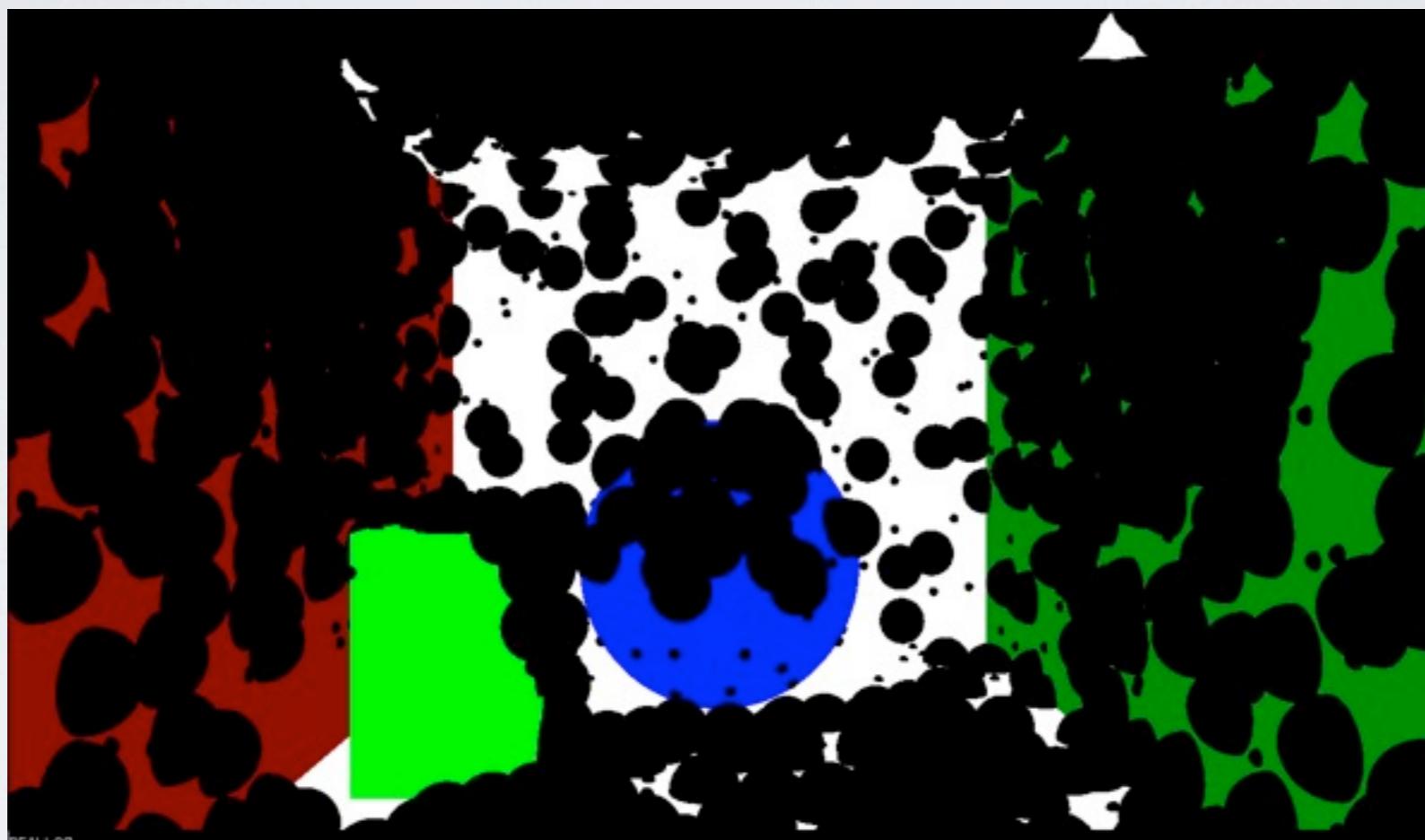
35GB de RAM, 55min sur corei7



4096 VPL (visibles)
avec lightbuffers (ombres)

INSTANT RADIOSITY

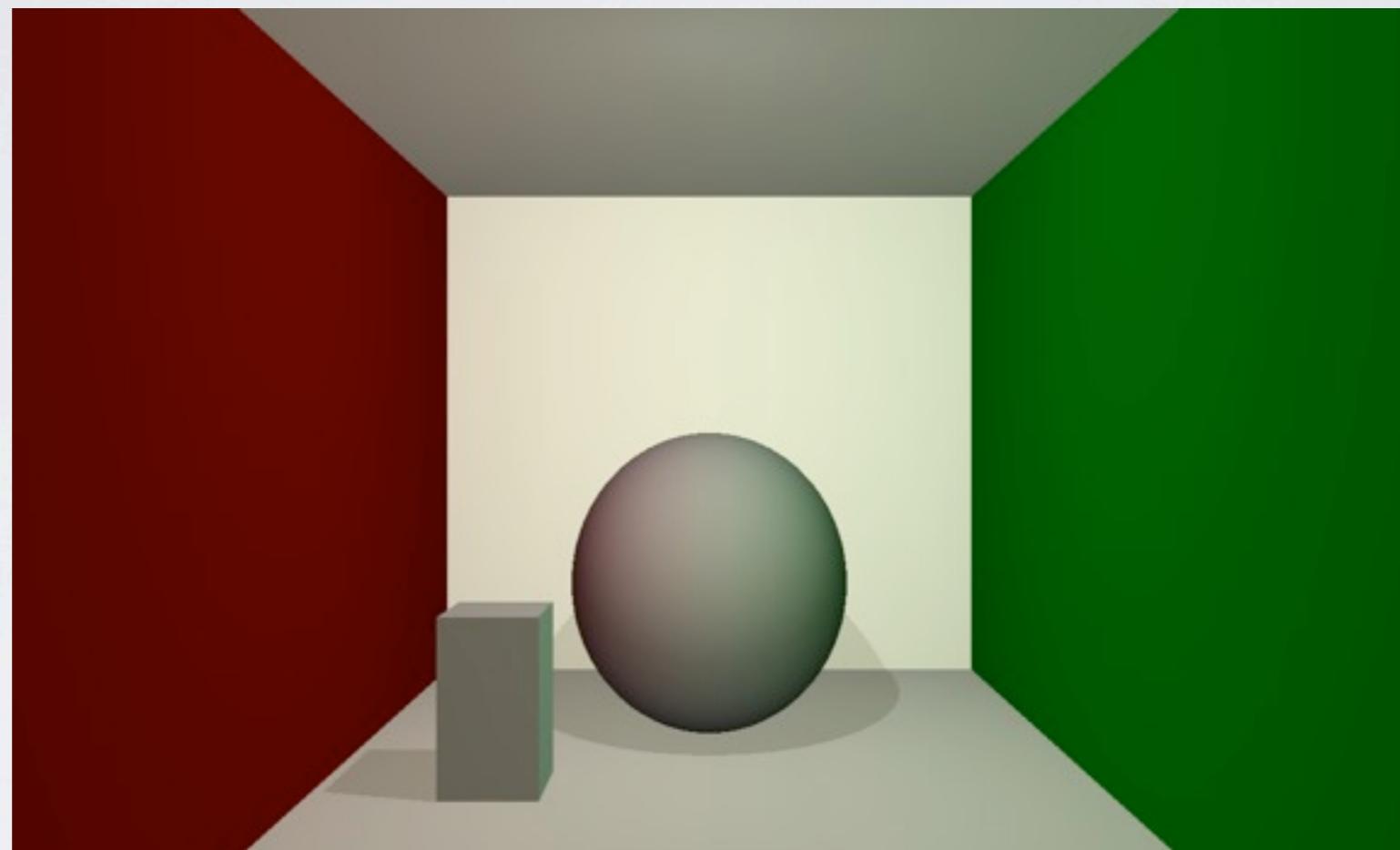
~15 minutes, RAM négligeable



2048 VPL (avec rebond)
position des VPL (sphères noires)

INSTANT RADIOSITY

rendu rapide, propre
quelques minutes, RAM négligeable



400 VPL (avec rebond)
une pointlight blanche

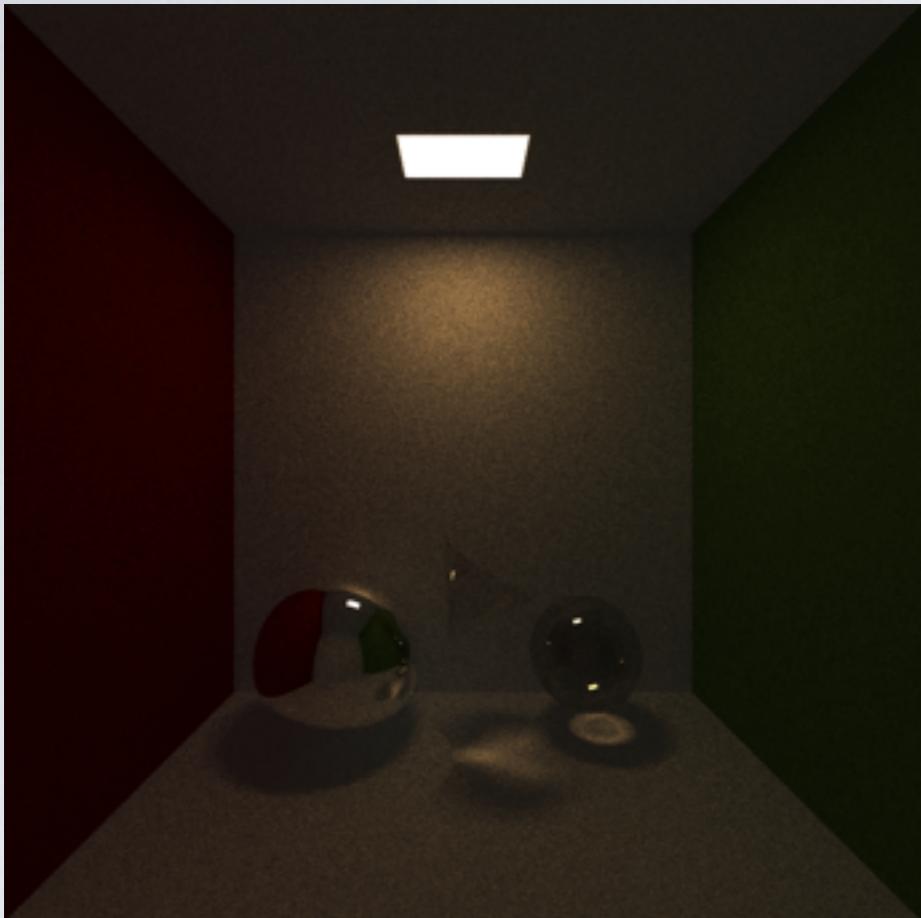
RENDERPASS

path tracing

PATH TRACING

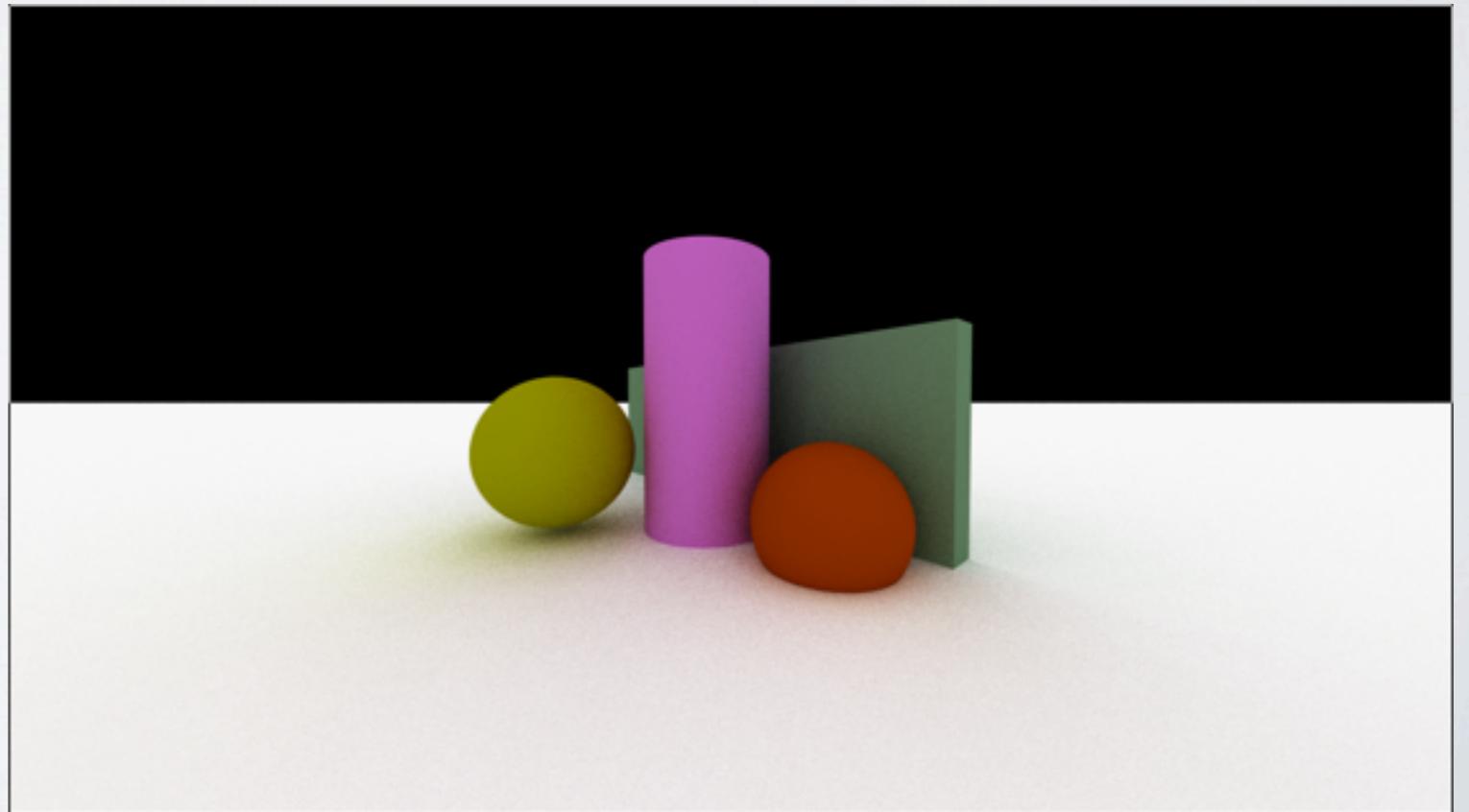
- renderpass complètement indépendante
- rendu parfaitement fidèle à la réalité
- n'est pas biaisé: permet un rendu photoréaliste si le nombre de samples est assez élevé
- utilisé pour générer des images de référence pour comparer d'autres techniques biaisées (par ex.: Photon Mapping)

PATH TRACING



Objets en “plein air”:
environment light uniquement

Cornell box avec caustiques:
sphère et cône transparent,
sphère réflechissante

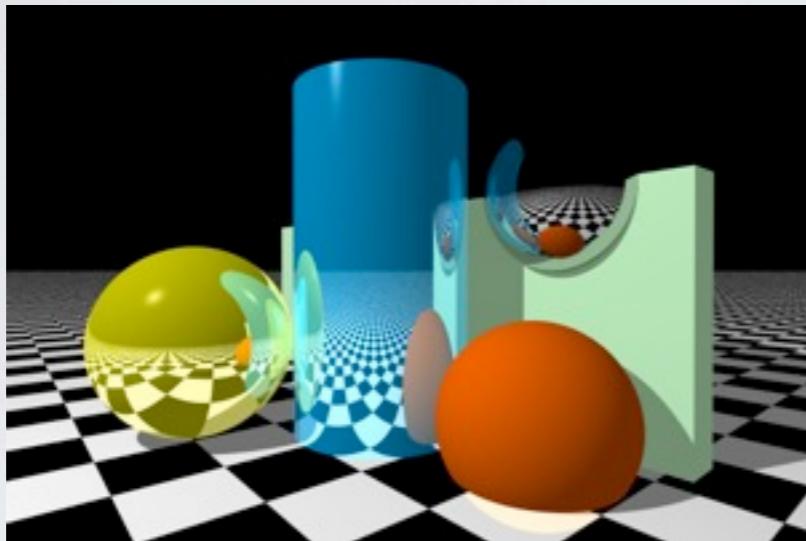


RENDERPASS

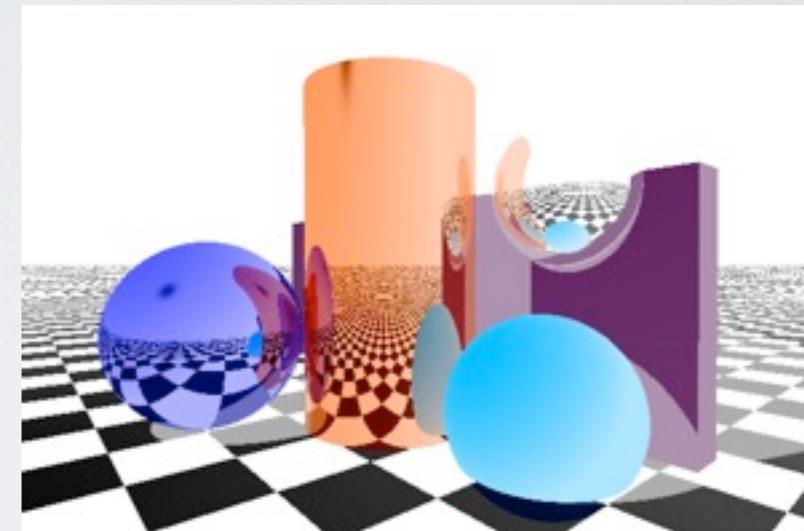
post processing

POST PROCESSING

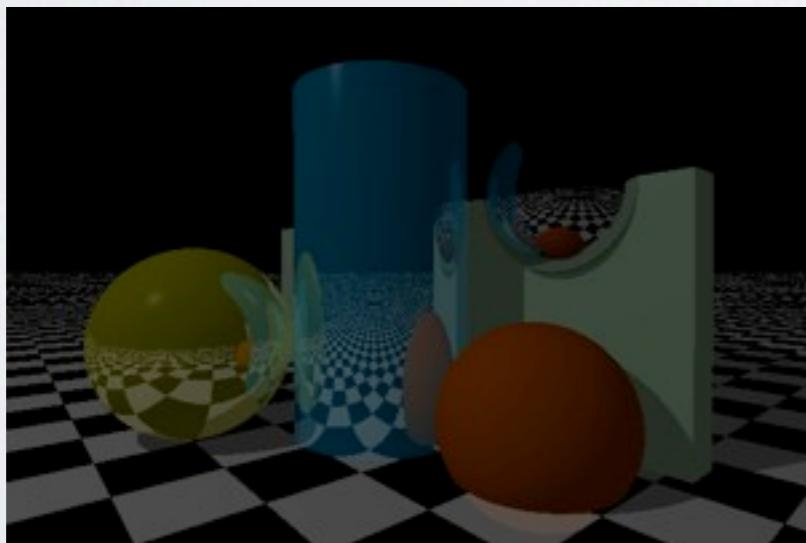
Original Image



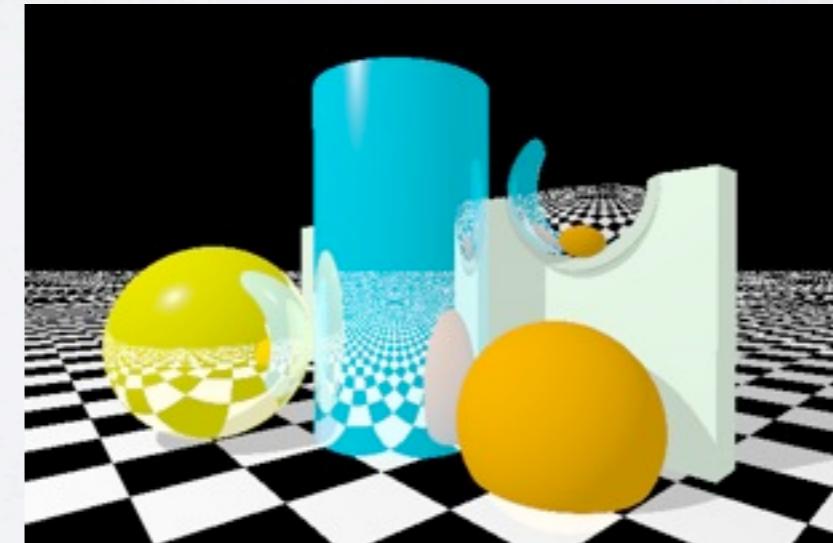
Negative Render Pass



ToneMapping Render Pass, coeff=5



Gamma Render Pass, e=3



OBJETS

"I raytrace objects as I think them, not as I see them." - PABLO PICASSO

OBJETS

plan

sphère

cylindre

cône

box

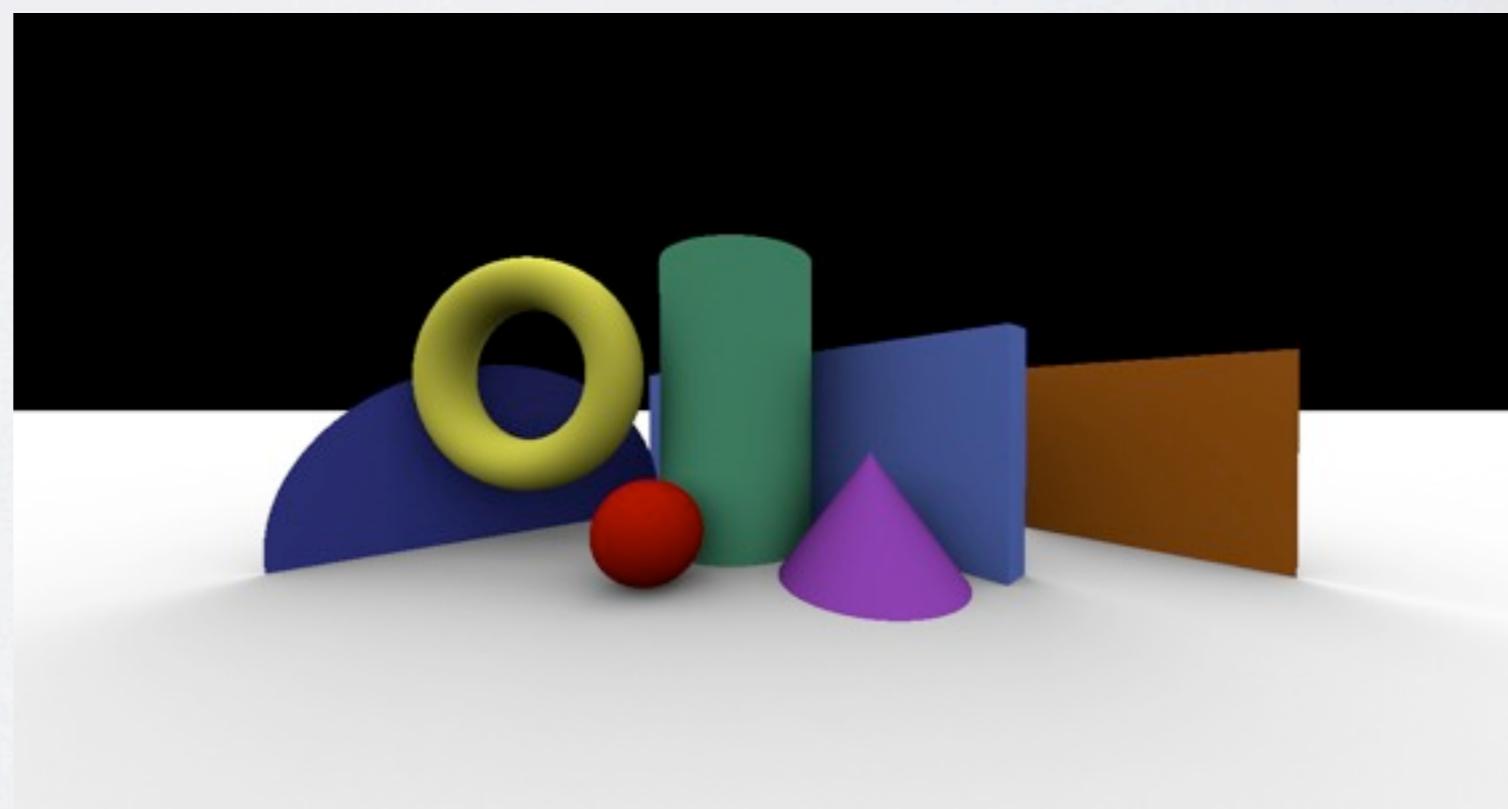
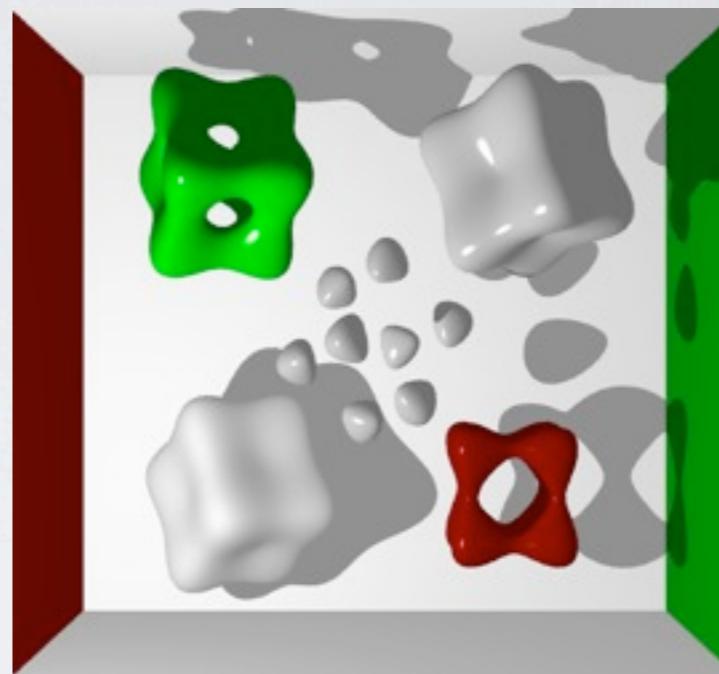
tore

rectangle

disque

cube troué

mesh



MESHES

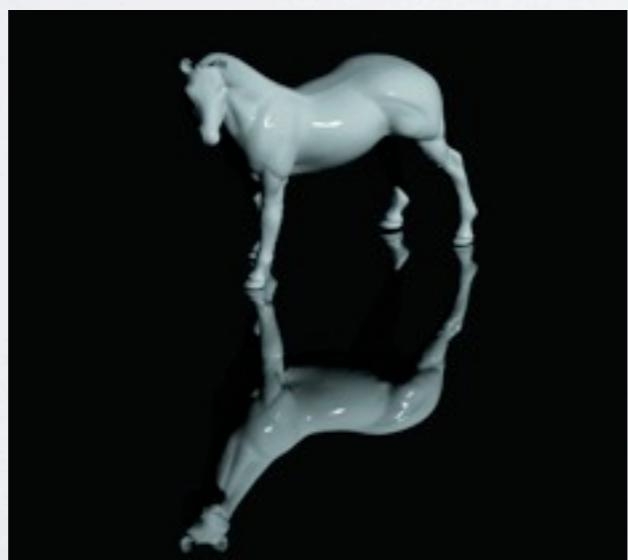
“Raytracing poneys is priceless” - MICKAEL RODRIGUES

MESHES

polygone composé de triangles

support des fichiers .ply

supporte tout les types de matériaux (comme les objets classiques)

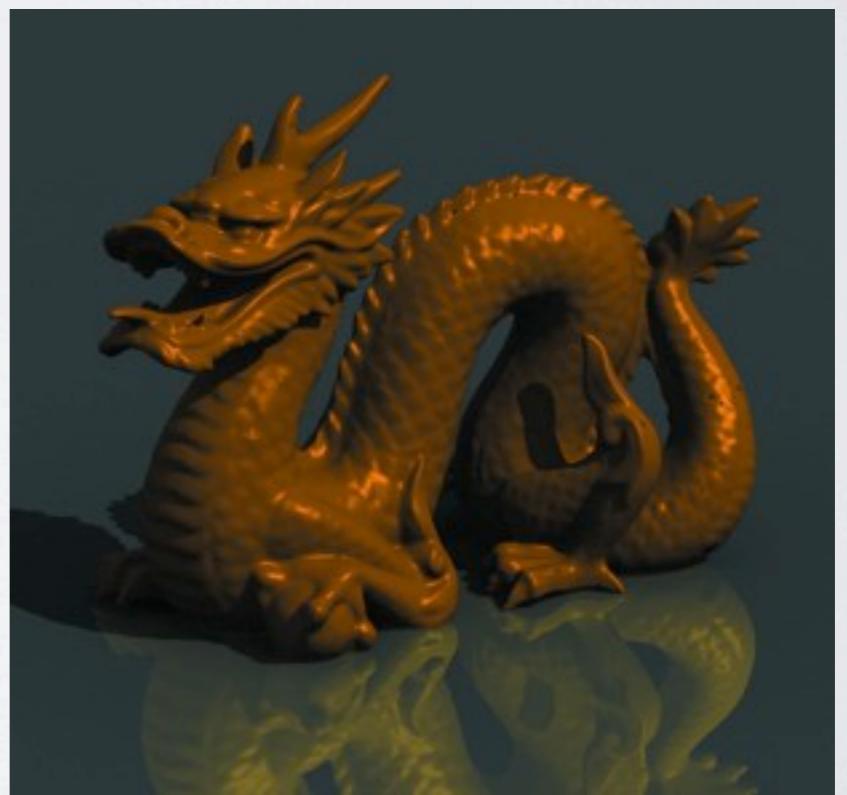


format ascii ou binaire

contient:

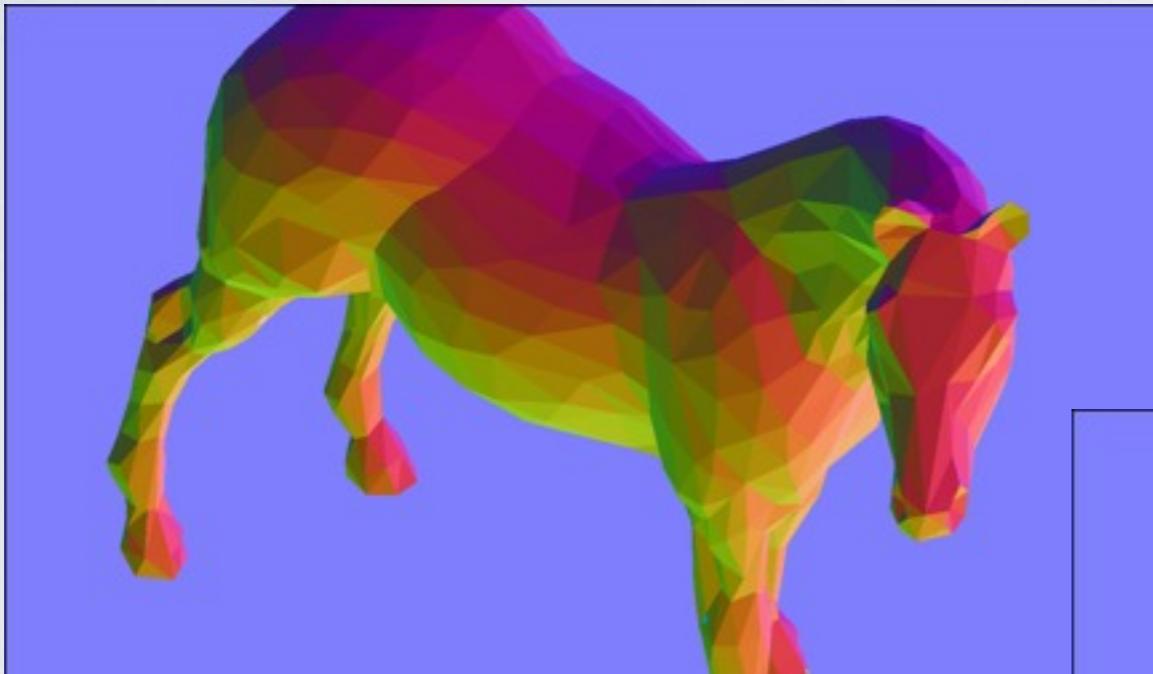
- liste de vertices (noeuds)
- liste de faces (ensemble de 3 vertices)

utilisation de ply.h/plyfile.c de Greg Turk
afin de lire le fichier

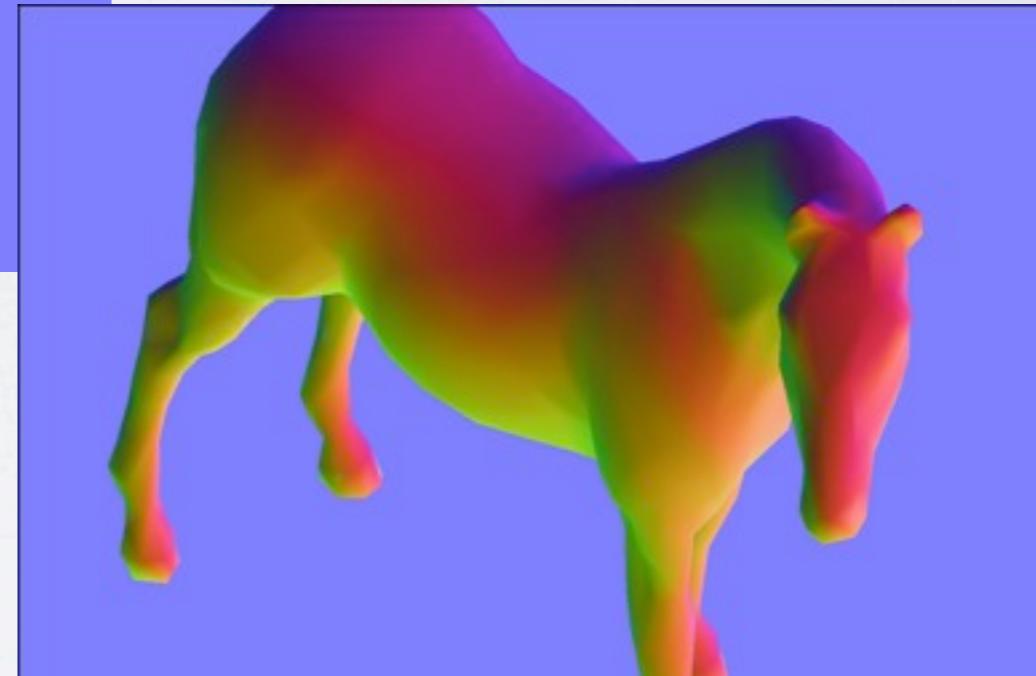


MESSES FLAT/SMOOTH

deux modes de rendu



flat

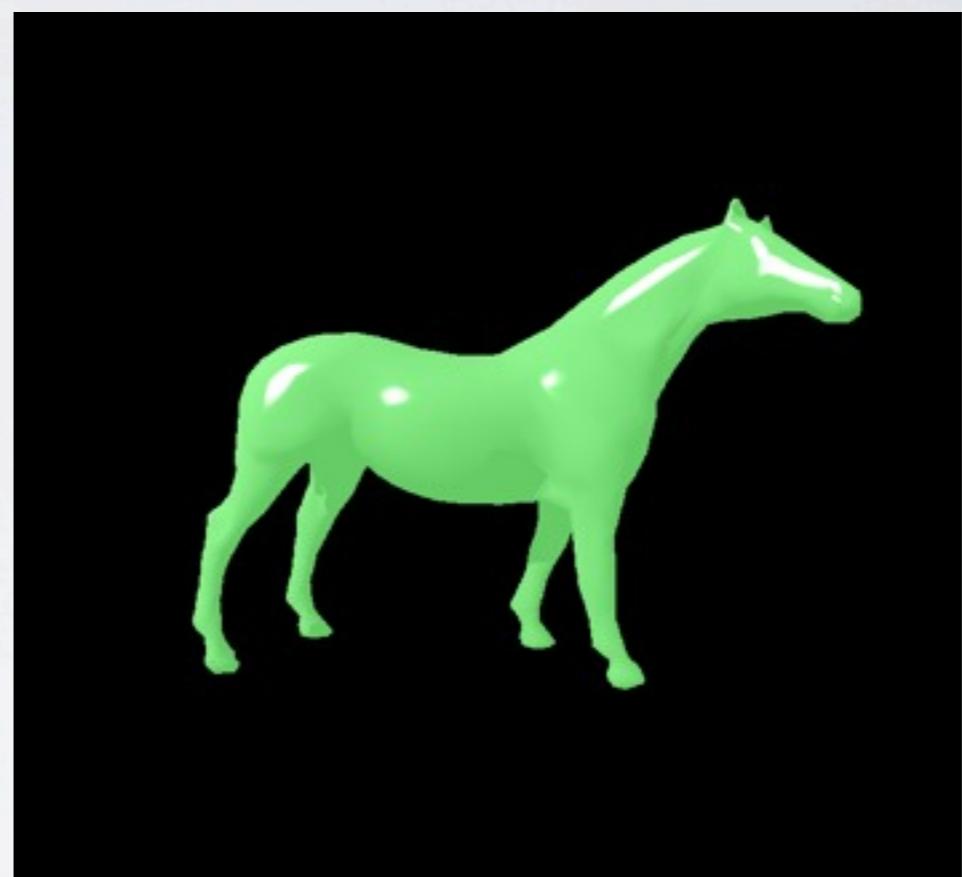
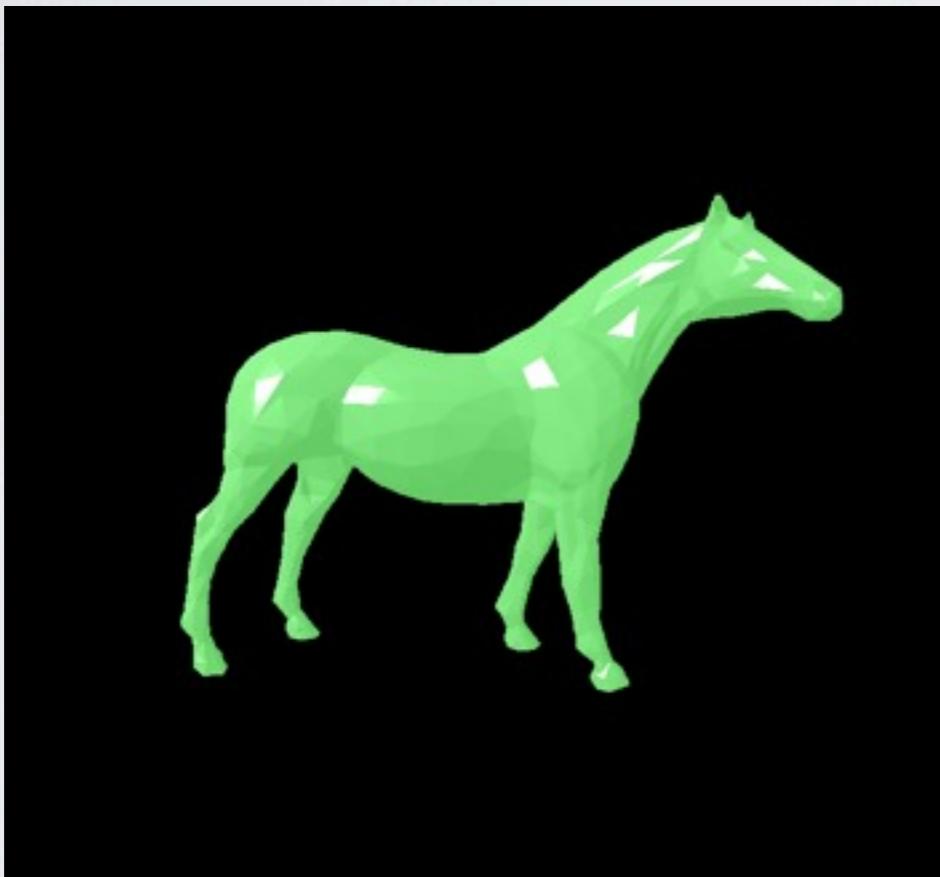


smooth

affichage du *normal buffer*

MESSES FLAT/SMOOTH

mise en évidence avec l'éclairage phong

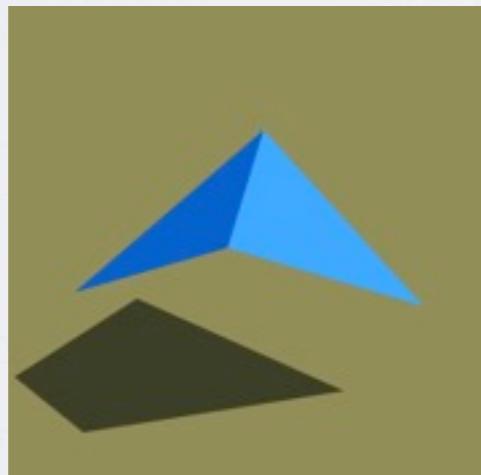


le rendu smooth permet de cacher un faible nombre de polygones mais cela reste visible sur les contours et les ombres

MESHS FLAT/SMOOTH

flat

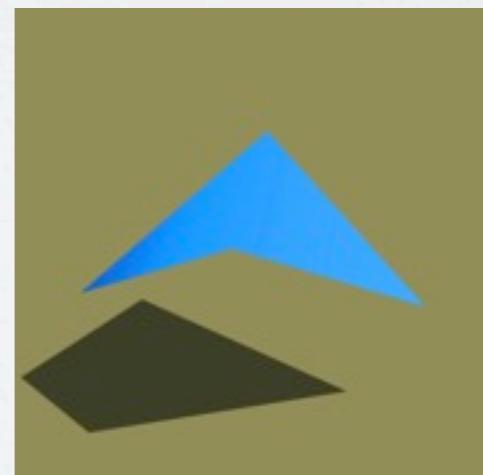
pour un triangle donné, la normale est la même en tout point d'intersection



smooth

une normale est calculée pour chaque triangle

à chaque intersection cette normale est interpolée (très peu de perte de performance)



MESSES GRID

structure de donnée 3d, division de l'espace

souvent le plus adapté pour les meshes (qui sont très denses)

permet de tester très peu d'intersections par rayon

de très nombreux objets peuvent donc etre raytracés sans perte de performance notable



MESSES GRID

nombres
de triangles

avec grid

sans grid

1 000

quelques secondes

plusieurs heures

100 000

quelques secondes

plusieurs jours

1 000 000

quelques secondes

impensable

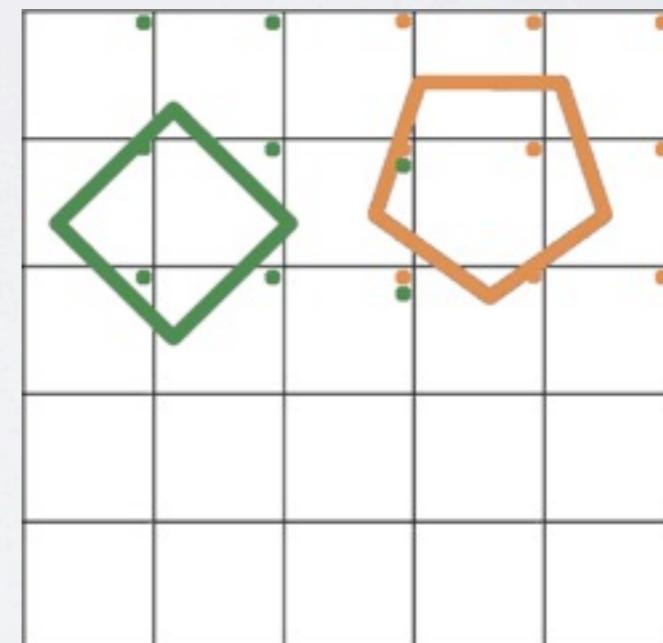
Initialisation

création d'une regular grid 3d couvrant la bounding box du modèle

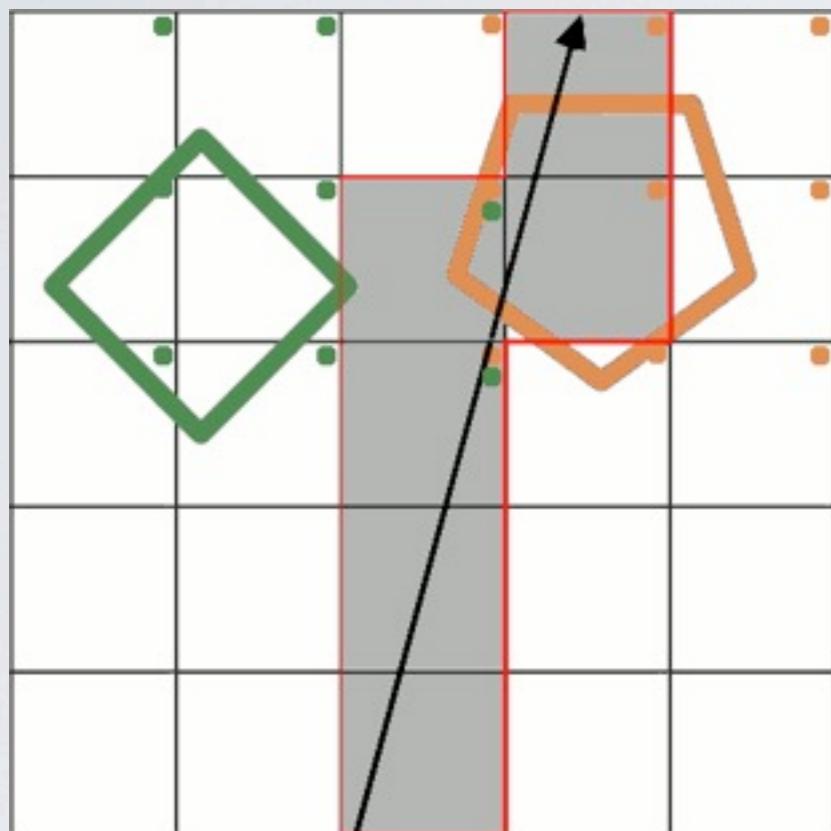
chaque case de la grid contient une liste d'objets

les triangles sont placés dans les cases qui intersectent avec la bounding box du triangle

vue en 2d



Parcours de la grid



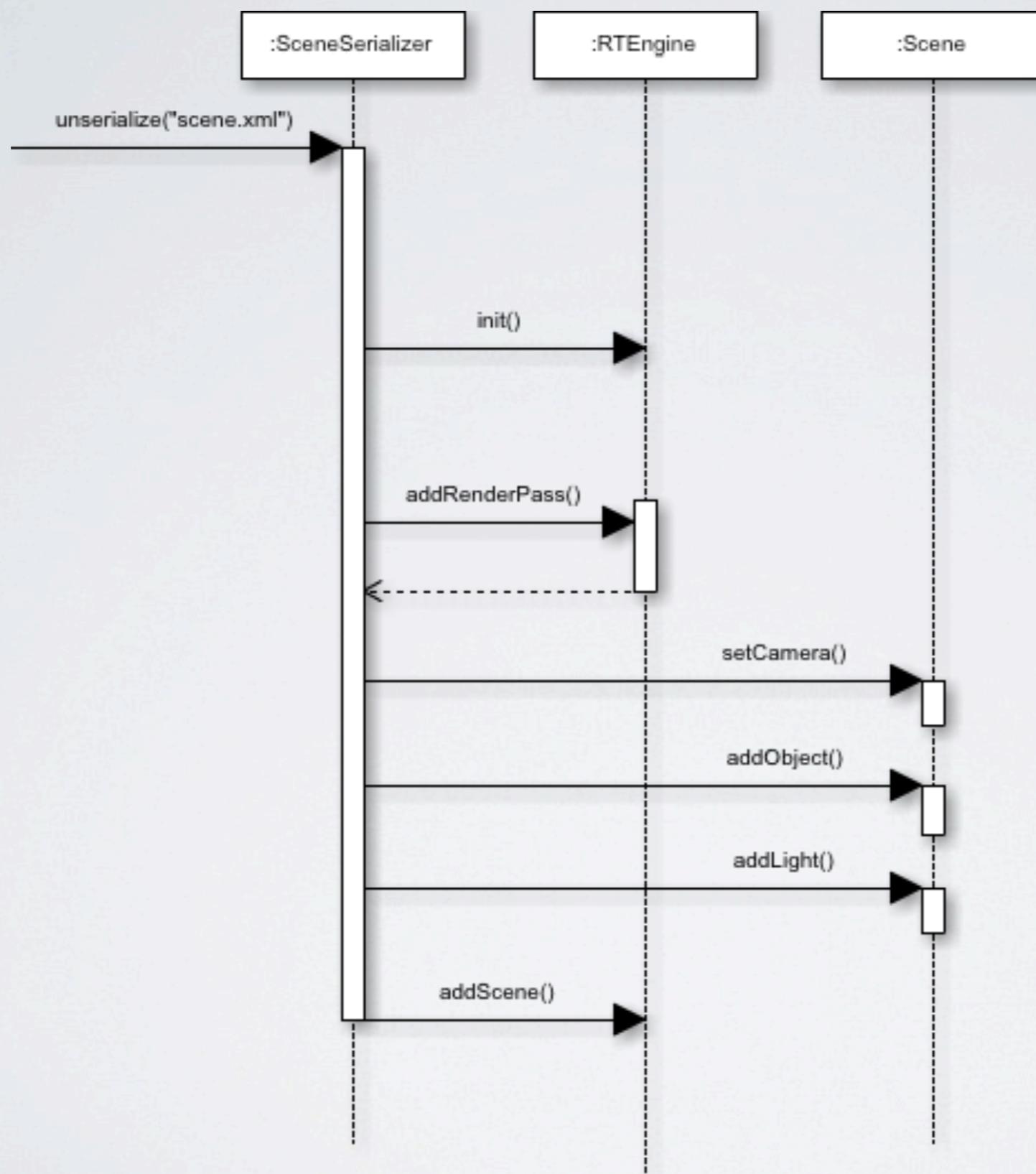
les calculs d'intersection sont seulement fait sur les objets voisins au rayon

plus la grid est dense, plus le nombre d'objets par case est faible et plus le parcours est rapide

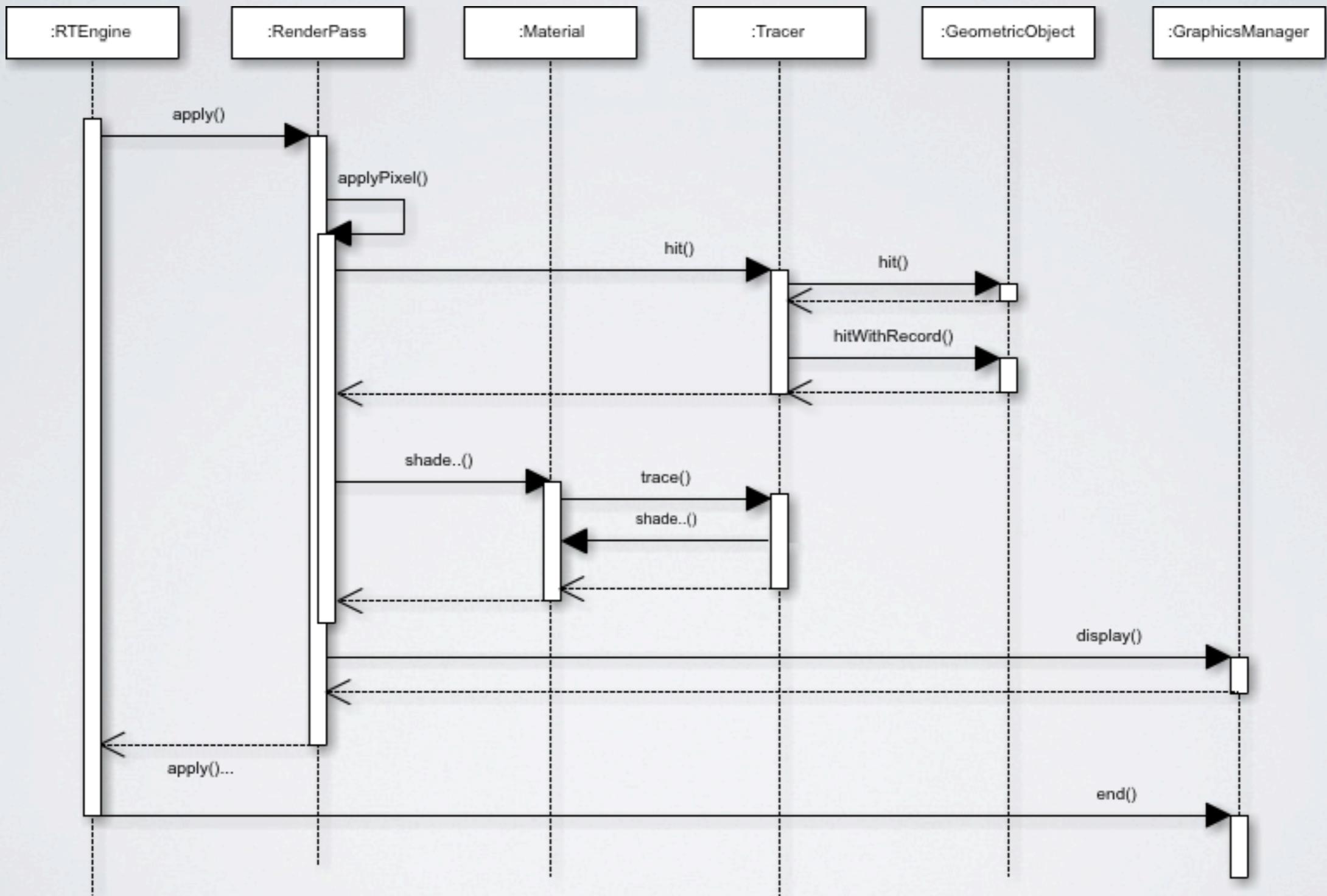
défaut : utilise beaucoup de mémoire

FONCTIONNEMENT

FONCTIONNEMENT



FONCTIONNEMENT



FONCTIONNEMENT

OOC

FONCTIONNEMENT OOC

Un raytracer se prête parfaitement au paradigme objet:

- Projet relativement large (30.000+ LOC)
- plusieurs groupes de “classes” similaires
(cameras: pinhole, spherical,.. ; objets: plan, sphere,.. ; renderpass, ...)
- fonctions qui n’ont de sens qu’avec l’état associé
(pour récupérer la direction de vision d’une camera, par exemple)
- l’encapsulation est indispensable
(à moins d’apprécier les forêts d’IFs)

FONCTIONNEMENT OOC

Impossibilité de coder en c++:

- modèle OOC de Schreiner
- permet d'éviter de dupliquer une grande partie du code
- permet d'encapsuler les différents types objets

Néanmoins, émuler une surcouche objet du C:

- requiert une très grande rigueur et discipline
- oblige à s'attarder sur les détails bas-niveau du c++ (RTTI,..)
- ajoute beaucoup de problématiques propres au C
- peut, si “mal” fait, amener à certaines choses très sales (typographie GTK, par exemple)

FONCTIONNEMENT OOC

code parfois très lourd:

- noms de fonctions à rallonge
- obligé de passer l'objet en premier argument
- impossibilité d'utiliser la surcharge d'opérateur: 8 lignes pour ce qui tiendrait en une ligne de c++
- warnings inadaptés
- ..

des solutions doivent être trouvées

PARALLELISATION

PARALLELISATION

multi-threading
clustering

PARALLELISATION

multi-threading

MULTI-THREADING

Utilisation d'OpenMP:

Portable: Industry Standard

Scalable: Répartition dynamique de la charge de travail

Facile à intégrer sur un travail existant

PARALLELISATION

clustering

CLUSTERING SERVEUR

Usage : ./rt scene.xml Cluster:server [Fork:yes|no]

affiche l'image

- fork un client
- envoie la scène aux clients
- répartit le travail entre les clients
- gère les {dé,re}connexions des clients

CLUSTERING CLIENT

Usage : ./rt Cluster:host

calcule l'image

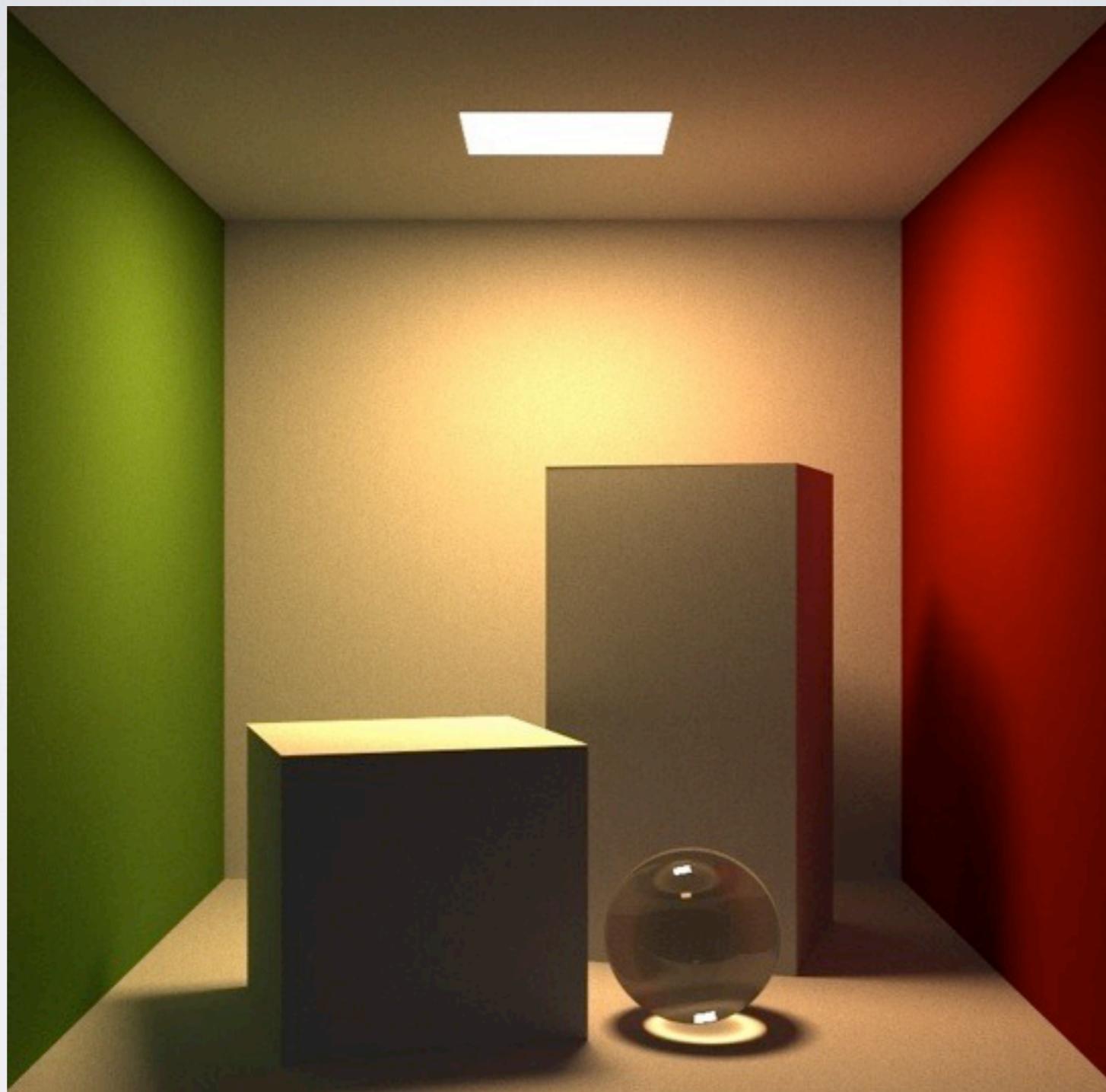
- reçoit le fichier de scène
- reçoit une colonne à calculer
- calcule en multi-threading
- affiche les colonnes calculées

RENDUS

RENDUS

128 min
43 pc (Lab Pasteur)

RENDUS



RENDUS

129 min

130 pc (Lab Pasteur; SM23, SM24)

RENDUS

