



ÉCOLE  
**CENTRALE**LYON

# ÉCOLE CENTRALE LYON

MOS 08 5  
RAPPORT

## Rendu du projet d'informatique graphique

*Élèves :*  
Tariq CHELLALI

*Enseignant :*  
Nicolas BONNEEL

## Table des matières

<b>1 Rendu final</b>	<b>2</b>
1.1 Salon aux couleurs vives . . . . .	2
<b>2 Temps d'exécution de chaque partie du Ray-tracing</b>	<b>3</b>
2.1 Sphère, Scene, ombre porté et correction Gamma . . . . .	4
2.2 Miroir, transparence, transmission de fresnel . . . . .	4
2.3 Eclairage indirecte . . . . .	5
2.4 Anti-aliasing . . . . .	5
2.5 Ombre douce et sphère émulsive . . . . .	6
2.6 Depth of field . . . . .	7
2.7 Maillage d'un objet avec la méthode de boîte englobante . . . . .	8
2.8 Lissage de phong . . . . .	8
2.9 Optimisation avec la méthode BVH et Textures . . . . .	9
2.10 Mouvement de la caméra . . . . .	10
2.11 Mappage de texture sur une sphère . . . . .	10
<b>3 Credit/Citation</b>	<b>11</b>

# 1 Rendu final

## 1.1 Salon aux couleurs vives

Pour le rendu final, j'ai décidé d'implémenté un rendu qui contient toutes les étapes du cours et qui "*se rapproche*" d'une image d'un salon avec des couleurs vives (*voir figure 1*) générée par l'IA. J'ai considéré cette image comme une image de référence pour s'amuser à implémenter le rendu finale mais j'ai ajouté des modifications de ma part à la limite des objets que j'ai trouvé sur le web.



FIGURE 1 – Image de référence générée par l'IA pour le rendu

L'implémentation de cette scéne nécessitait l'importation de differents fichier .obj.

- **Un chat** qu'on peut télécharger [ici](#) .
- **Un canapé** qu'on peut télécharger [ici](#) .
- **Une table** qu'on peut télécharger [ici](#) .
- **Un tapis** qu'on peut télécharger [ici](#) .
- **Un pot** qu'on peut télécharger [ici](#) .

Le temps d'exécution de l'image dans la figure 2 (**Figure 2**) pour **1000 rayons/pixel** est de **4188.81 secondes**.

Le temps d'exécution de l'image du rendu final avec une sphère transparente (**Figure 3**) de taille **1204\*768** avec **1500 rayon/pixel** est de **16178.3 secondes**.

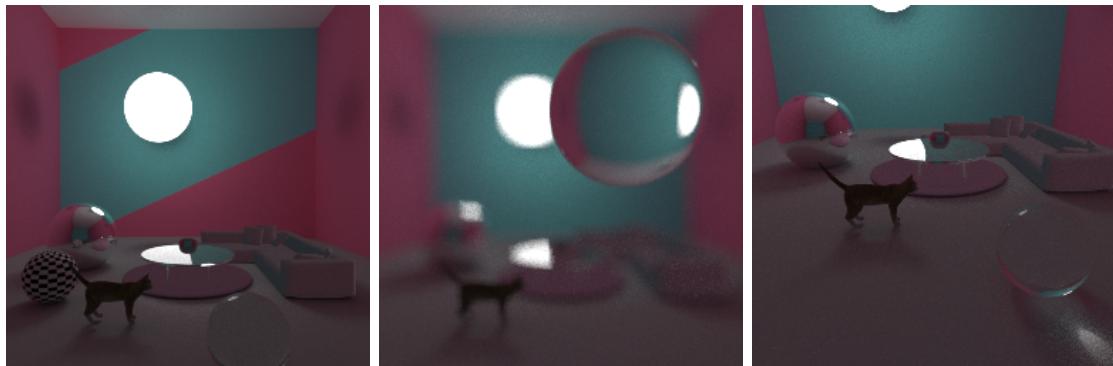


FIGURE 2 – Image du rendu final avec 1000 rayon/pixel (**à droite**) image du rendu final avec profondeur de champs avec 400 rayon/pixel (**au milieu**) rendu final avec 400 rayon/pixel avec mouvement de la caméra (**à gauche**)

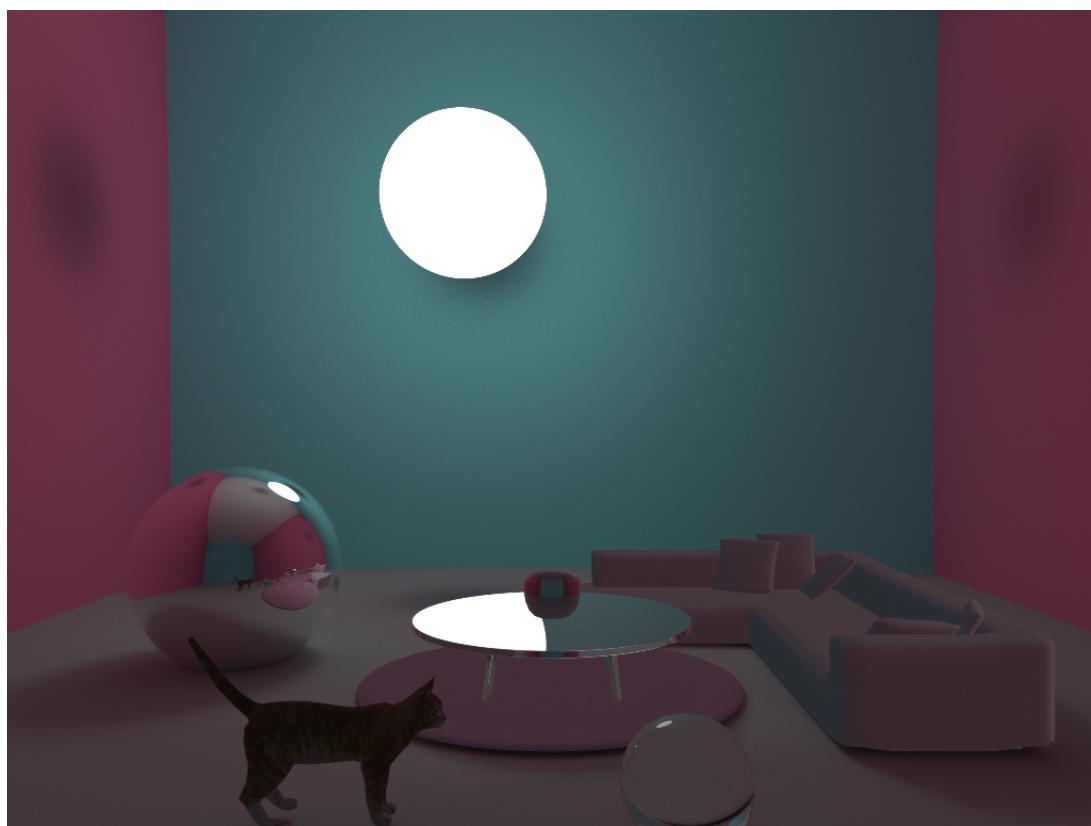


FIGURE 3 – Image du rendu final de taille (1204\*768) avec 1500 rayon/pixel

## 2 Temps d'exécution de chaque partie du Ray-tracing

Dans cette section, je décrirais chaque étape importante de construction du rendu final en explicitant le temps d'exécution pour générer chaque image. Mes notes de cours

"Note\_de\_cours.md"

construit à orienter ces rendus sont disponible dans le repo Github *Raytracing-with-C*.

## 2.1 Sphère, Scene, ombre porté et correction Gamma

Le temps d'exécution de chaque partie (en taille 512\*512) est résumé dans le tableau suivant :

Partie	Temps d'exécution en seconde
Scène	0.152
Ombre porté	0.237
Correction Gamma	0.234

Les résultats de chaque partie (en taille 512\*512) sont affiché dans *la figure 4* ci-dessous :

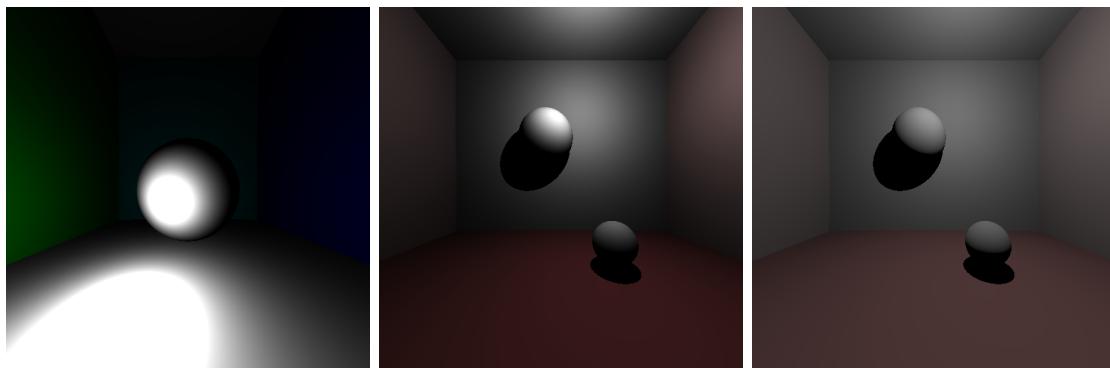


FIGURE 4 – Rendu de la scène (**à gauche**), rendu des ombres porté (**au milieu**) rendu de la scéne avec la correction Gamma (**à droite**).

## 2.2 Miroir, transparence, transmission de fresnel

Le temps d'exécution de chaque partie (en taille 512\*512) est résumé dans le tableau suivant :

Partie	Temps d'exécution en seconde
Miroire	0.249
transparence	0.233
transmission de fresnel	0.296

Les résultats de chaque partie (**512\*512**) sont affiché dans *la figure 5*.

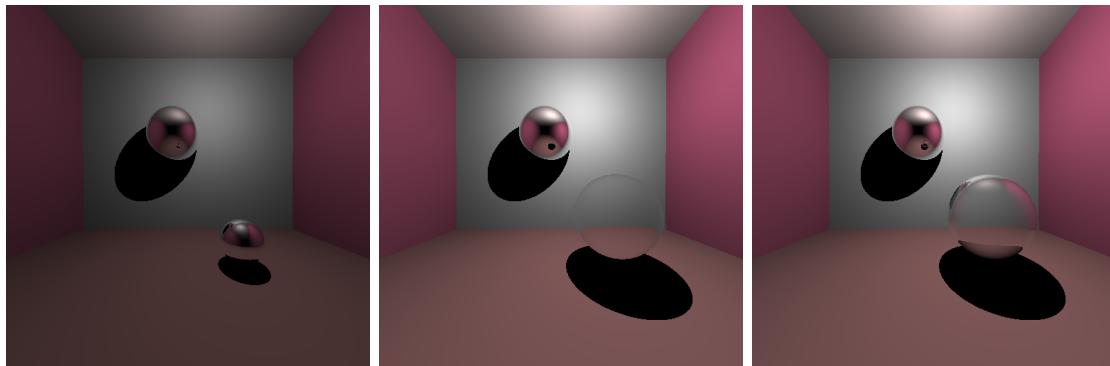


FIGURE 5 – Rendu du Miroir (**à gauche**), rendu de transparence (**au milieu**) rendu de transmission de fresnel (**à droite**).

### 2.3 Eclairage indirecte

Le temps d'exécution par nombre d'échantillons avec **un nombre de rebond fixée à 5** et des images en taille **512\*512** est résumé dans le tableau suivant :

Nombre de rayons	Temps d'exécution en seconde
1	1.367
10	18.602
100	200.267
380	746.043

Les résultats de chaque partie (**512\*512**) sont affiché dans *la figure 6* :

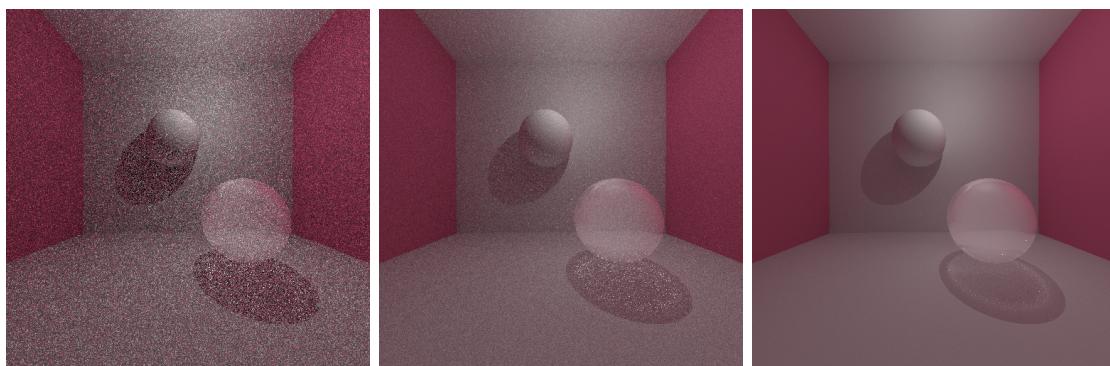


FIGURE 6 – Rendu de l'éclairage indirecte 1 rayon (**à gauche**), 10 rayons (**à gauche**), 380 rayons (**à droite**).

### 2.4 Anti-aliasing

Le temps d'exécution de l'anti-aliasing pour **un nombre d'échantillons fixé à 80** avec **un nombre de rebond fixé à 5** et des images en taille **512\*512** est résumé dans le tableau suivant :

Partie	Temps d'exécution en seconde
Aliasing	182.817
Anti-Aliasing	340.561

Les résultats de chaque partie (**512\*512**) sont affiché dans la figure 7 .

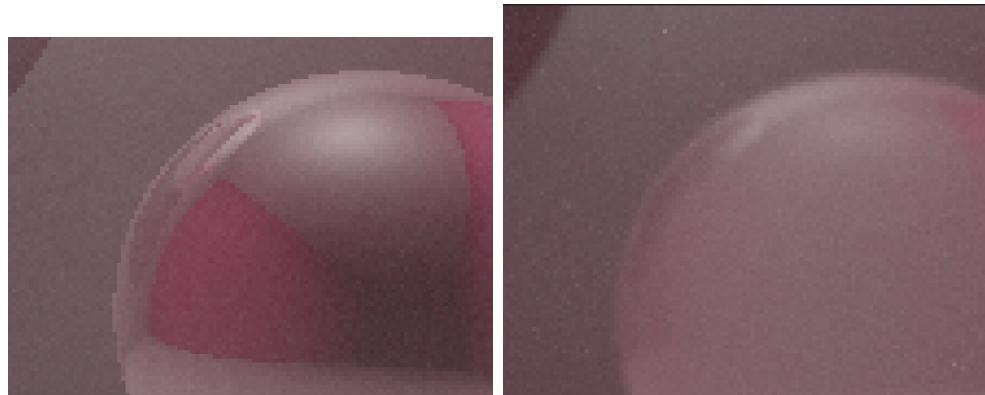


FIGURE 7 – Rendu de l'elairage indirecte avec Aliasing 80 rayon (**à gauche**), sans problème de Aliasing (**à droite**).

## 2.5 Ombre douce et sphère émersive

Je me suis rendu compte que j'avais quelques bug sur la fonction récursive *getColor*. Plus précisement, dans la partie du calcul de l'eclairage indirecte, transparence et la facon par laquelle j'utilisais la condition d'ombre. En plus, avec quelques optimisation de code et en corrigeant la prallélisation, j'ai réussi à diminuer largement le temps de calcul de chaque rendu.

Le temps d'exécution des ombres douces par nombre d'echantillons avec un nombre de **rebond fixé à 5** et des images en taille **512\*512** est résumé dans le tableau suivant :

Nombre de rayons	Temps d'exécution en seconde
1	0.181
10	0.396
380	19.327

Les résultats pour differents nombres de rayons/pixel avec des images de taille (**556\*556**) sont affiché dans *la figure 8*.

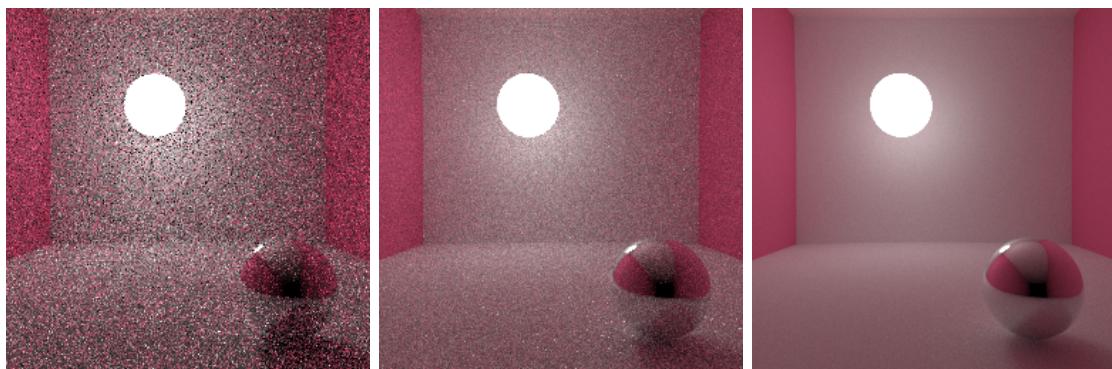


FIGURE 8 – Rendu des ombres douces 1 rayon (**à gauche**), 10 rayons (**au milieu**), 380 rayons (**à droite**).

## 2.6 Depth of field

Le temps d'exécution de la profondeur de champ avec deux sphère une miroir et l'autre transparente pour un **nombre d'échantillons fixé à 180** avec un **nombre de rebond fixé à 5** et des images en taille 512\*512 est de **16.365 secondes**. Le résultat de cette partie est affiché dans *la figure 9*.

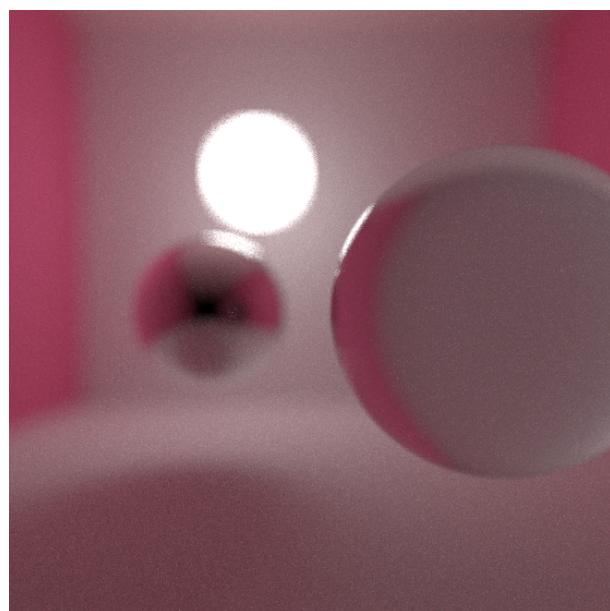


FIGURE 9 – Rendu de la profondeur de champs avec 180 rayons .

## 2.7 Maillage d'un objet avec la méthode de boîte englobante

Le temps d'exécution du maillage en utilisant la méthode de la boîte englobante de différent objets :

- Un chat qu'on peut télécharger [d'ici](#).
- Une table qu'on peut télécharger [d'ici](#).
- Une chaise qu'on peut télécharger [d'ici](#).

Cela pour un **nombre d'échantillons fixé à 180** avec un **nombre de rebond fixé à 5** et des images en taille **256\*256** décrit pour différents objets dans la scènes est résumé dans le tableau suivant :

Objets	Temps d'exécution en seconde
Chat + sphère miroir	52.893
Chat + Table + Sphère miroir	49.13

Les résultats de chaque partie sont affichés dans *la figure 10*.

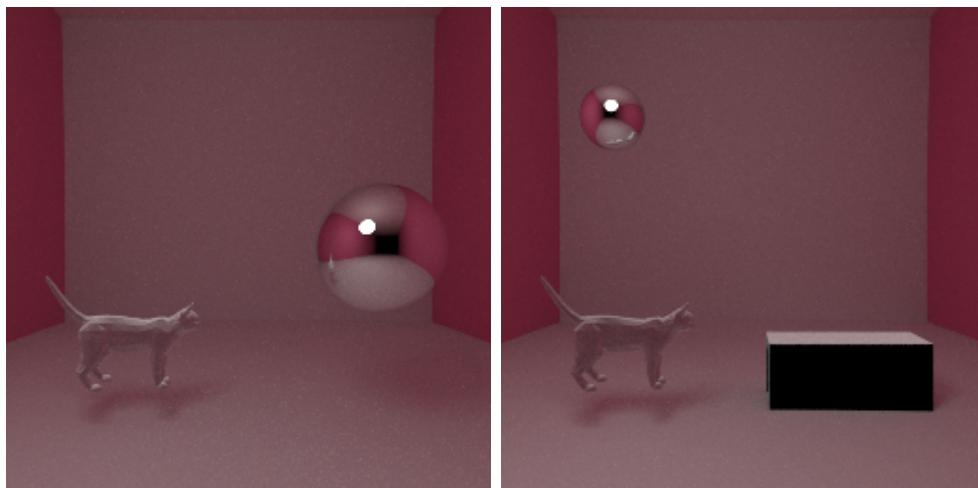


FIGURE 10 – Rendu du maillage du Chat + sphère miroire (à gauche), Chat + Table + Sphère miroir (à droite).

## 2.8 Lissage de phong

Le temps d'exécution du lissage pour un **nombre d'échantillons fixé à 180** avec un **nombre de rebond fixé à 5** et des images en taille 256\*256 est de **56.365 secondes**. Le résultat de cette partie est affiché dans *la figure 11*.

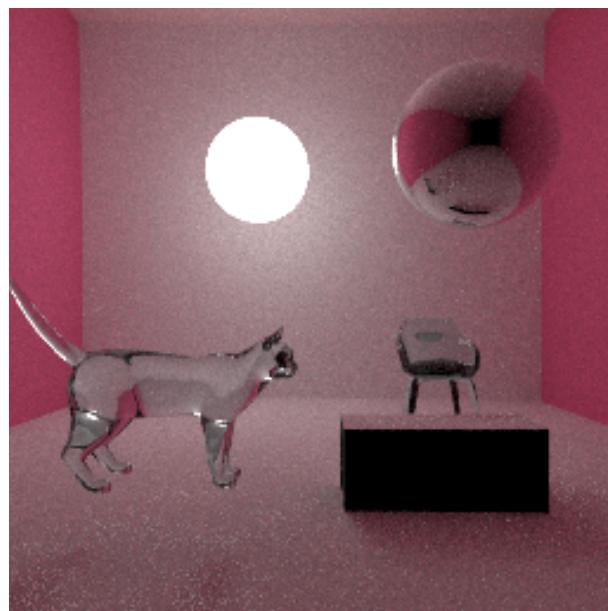


FIGURE 11 – Rendu du maillage avec lissage des normaux avec 180 rayons .

## 2.9 Optimisation avec la méthode BVH et Textures

Dans cette sections nous allons comparer le temps d'exécution de la méthode d'intersection avec la boite englobante avec la méthode d'optimisation bounding volume hierarchies (BVH) avec une scéne composée d'un chat et une sphére miroir.

Le temps d'exécution pour un nombre d'échantillons fixé à 380 et avec un nombre de rebond fixé à 5 et des images en taille 256\*256 est résumé dans le tableau suivant :

Objets	Méthodes ajoutées	Temps d'exécution en secondes
Chat + sphère miroir	bounding box simple	<b>376.593</b>
Chat + sphère miroir	lissage de Phong+ texture+ BVH	<b>287.624</b>

Le résultat de cette partie est affiché dans *la figure 12*.

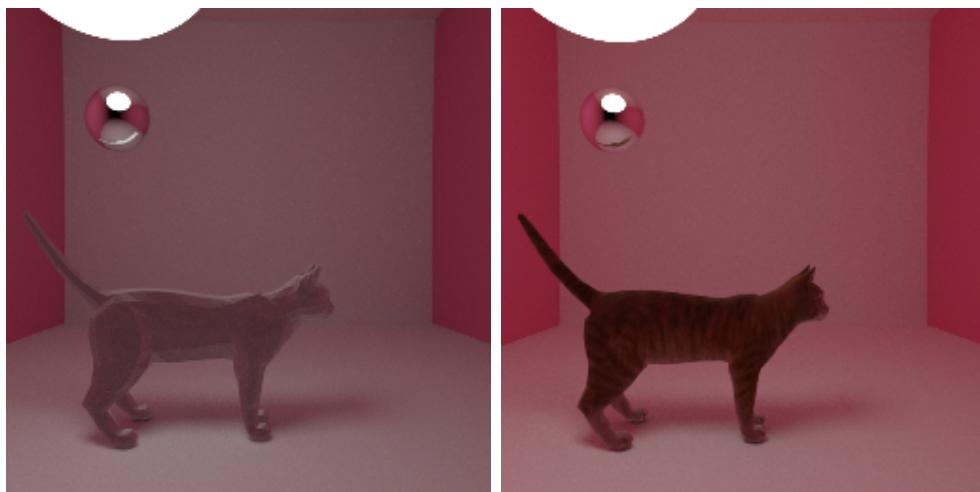


FIGURE 12 – rendu des maillage Chat avec méthode bounding box (**à gauche**), et Chat + texture + lissage de phong + sphère miroir + méthode BVH (**à droite**).

## 2.10 Mouvement de la caméra

Le temps d'exécution du rendu finale avec un mouvement de la caméra de 30° verticalement vers le bas et 5° horizontalement vers la droite est **411.416 secondes**. Le résultat de cette partie est affiché dans *la figure 13*.

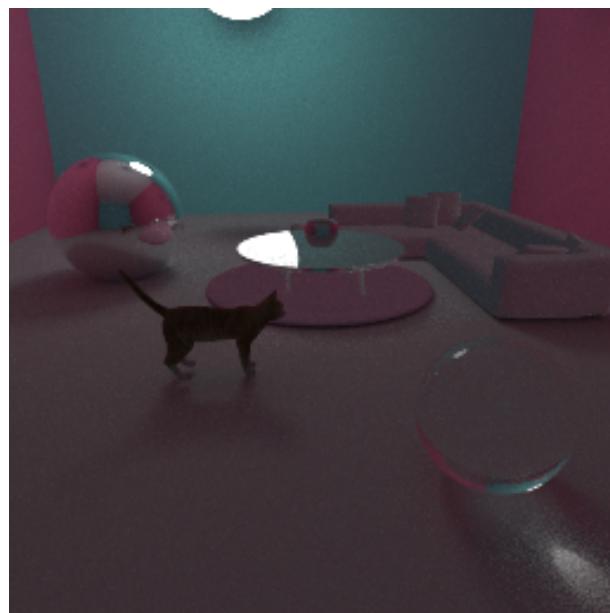


FIGURE 13 – Rendu finale avec mouvement de la caméra de 30° verticalement vers le bas et 5° horizontalement vers la droite

## 2.11 Mappage de texture sur une sphère

Le temps d'exécution du mapping carré avec un nombre de rayons par pixel fixé à 500 et une image de taille 256\*256 est de 37.195 secondes. Le résultats de cette partie sont affichés dans *la figure 14*.

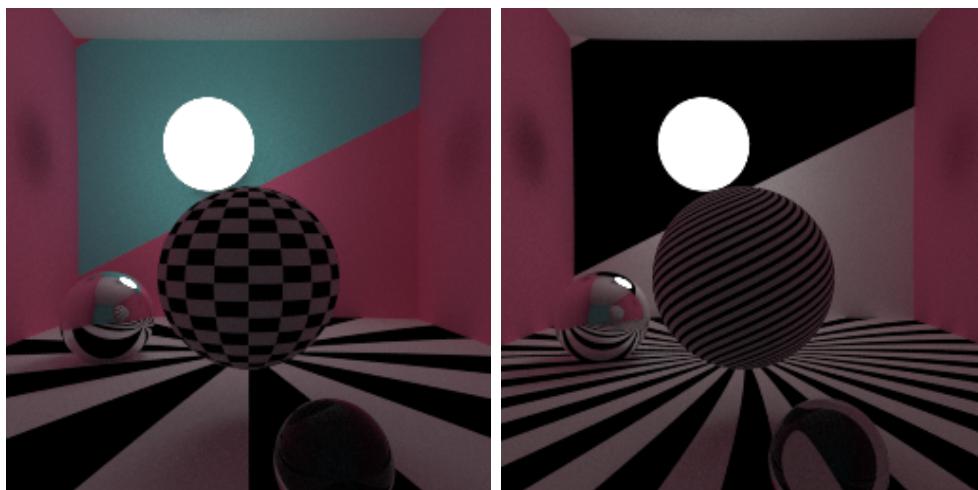


FIGURE 14 – rendu des mapping avec des carrés sur une sphère (**à gauche**), et rendu des mapping avec des cercles sur une sphère (**à droite**).

### 3 Credit/Citation

Ce travail s'appuie sur quelques ressources qui méritent d'être reconnues :

- Le polycopié du cours de monsieur Bonneel disponible sur *ce lien*.
- Les vidéos du cours qui ont été utilisées comme support pour la compréhension des concepts approfondis comme l'éclairage direct et indirect. Ces vidéos sont disponibles sur *ce lien*.
- Une ressource externe que j'ai trouvée sur internet pour comprendre plus facilement quelques concepts sur la théorie du ray tracing disponible sur *ce lien*.
- Une ressource externe qui m'a aidé à comprendre le mappage sur des sphères disponible sur *ce lien*.