

## Rapport Indromatique Graphique

Remarque Préliminaire : Les temps donnés sont donnés en temps CPU car ils ont été obtenus avec la bibliothèque ctime qui travaille en temps CPU (à savoir que si votre PC a  $n$  coeurs, alors le temps d'exécution sera en principe divisé par  $n$ , puisque la plupart a été parrallélisé).

### 1 - L'étude d'intersections entre rayons et objets.

Une étude sur quel est le point d'intersection entre des objets géométriques, et les rayons envoyés par une camera permet de réaliser un affichage assez réaliste, bien qu'on constate des ombres peu réalistes puisque tout à fait noires :

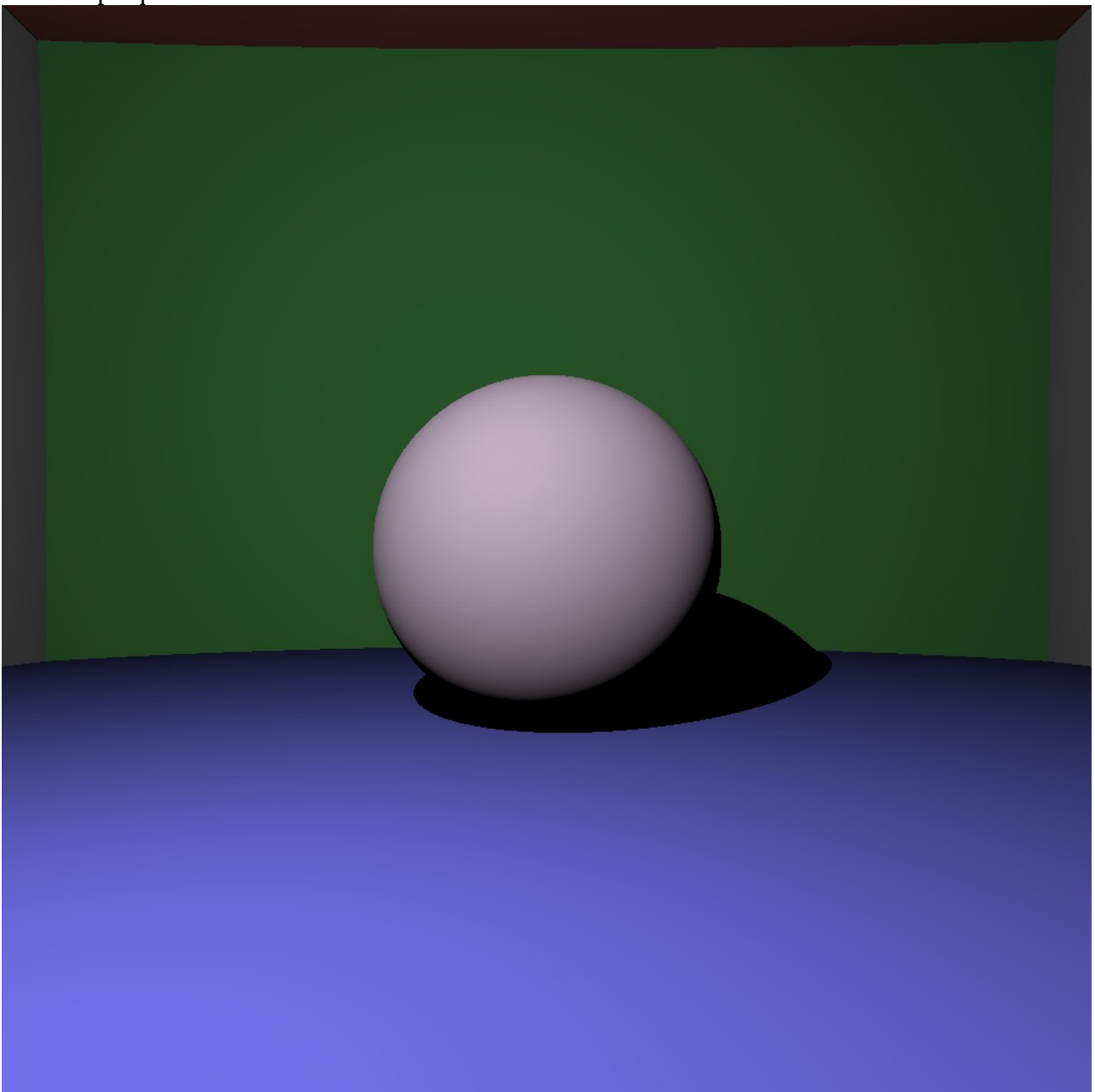


Image réalisée en moins d'une seconde.

2 - Modèles de BRDF pour faire ressortir des ombres douces.

L'équation du rendu est la suivante :

$$L_o(x, o) = E(x, o) + \int f(i, o) L_i(x, i) i \, di$$

En introduisant l'équation du rendu ainsi que certains modèles de BRDF, et en intégrant grâce à des méthodes de type Montecarlo, on est capables de produire des images avec des ombres douces :

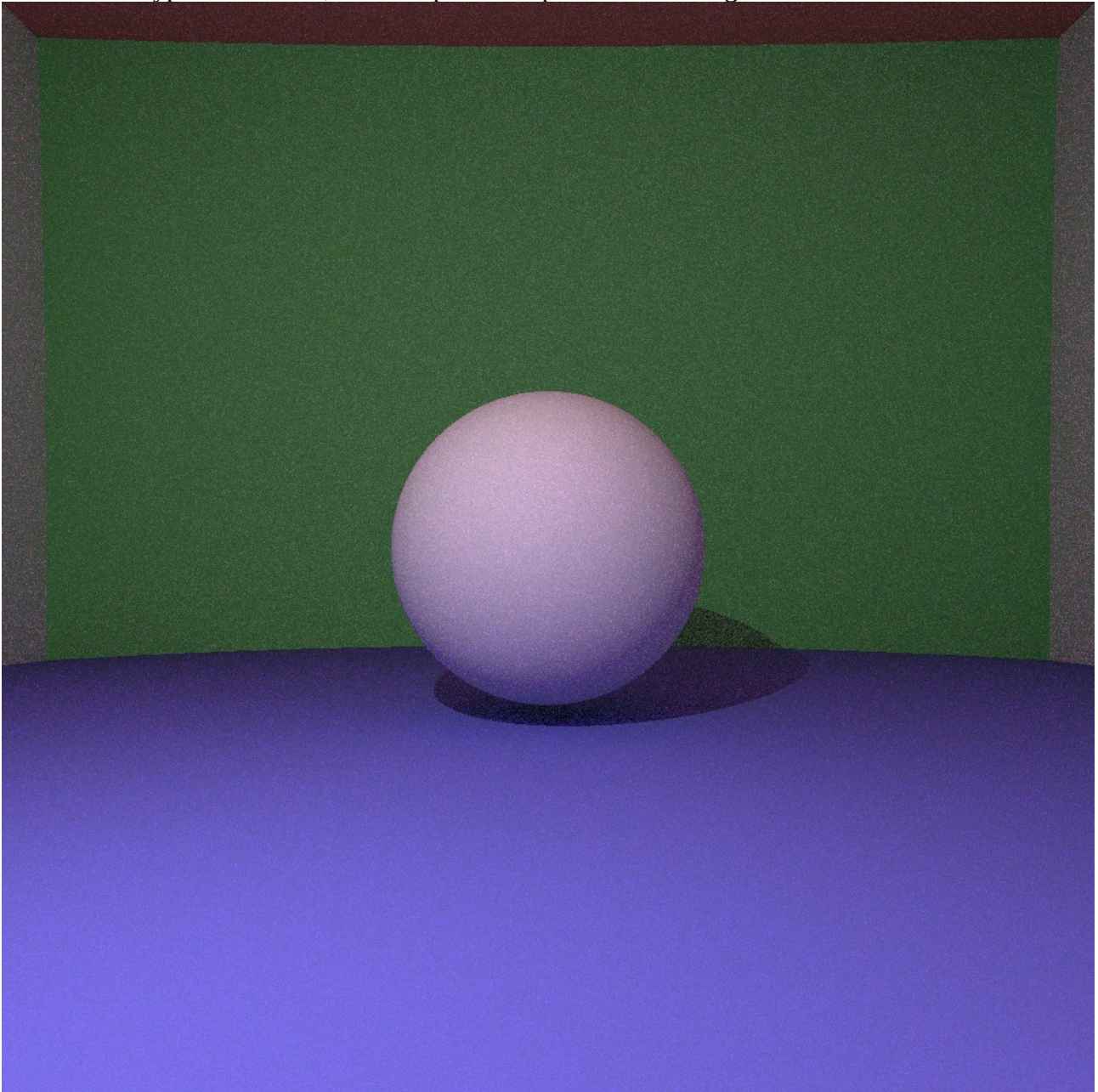


Image obtenue en 86 secondes temps CPU

Cependant, un zoom sur les bords de notre image montre un phénomène de pixelisation. Pour palier à ce problème, on réalise un antialiasing qui consiste à lancer de différents rayons en faisant varier légèrement l'orientation du rayon avant de le lancer :

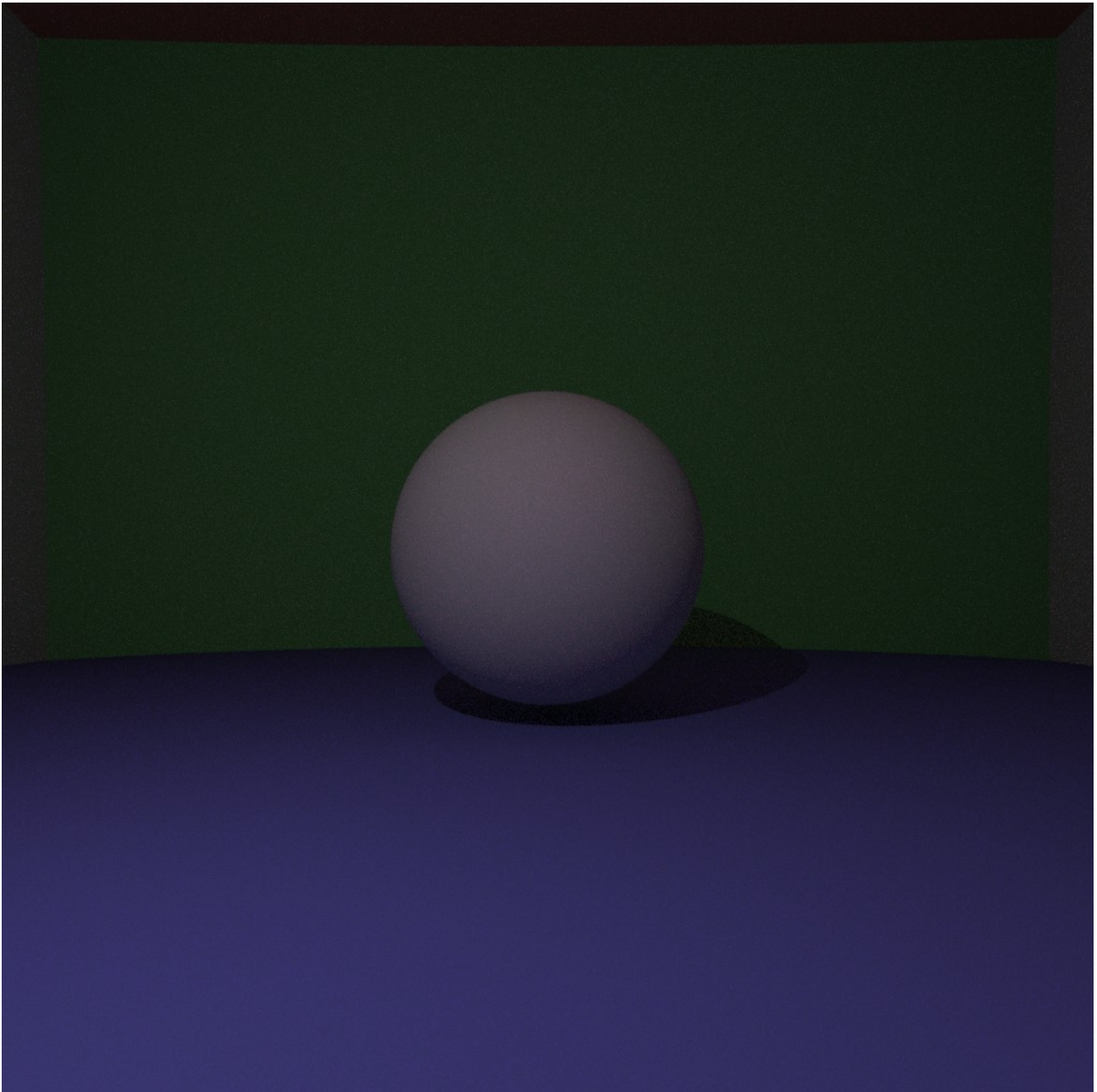


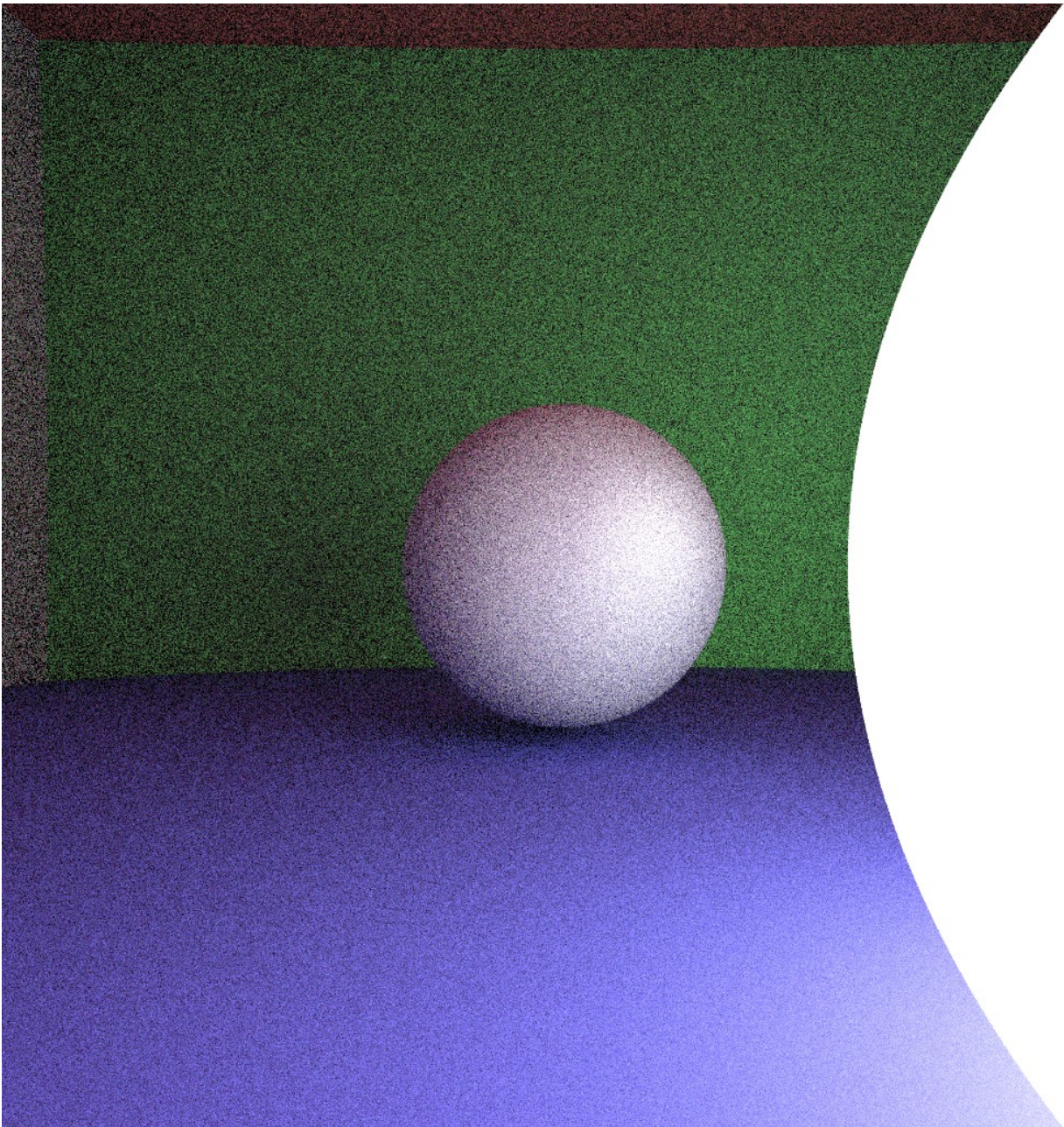
Image avec antialiasing obtenue en 63 secondes (temps CPU).

### 3 - Modèles de source lumineuse sphérique

Les rendus précédents ont été réalisés avec une source lumineuse ponctuelle. Or à part pour le soleil, on rencontre rarement ce genre de sources lumineuses. Ainsi, on ajoute une source lumineuse sphérique.

Dans un premier temps, on essaye d'obtenir des images uniquement avec l'éclairage indirect :

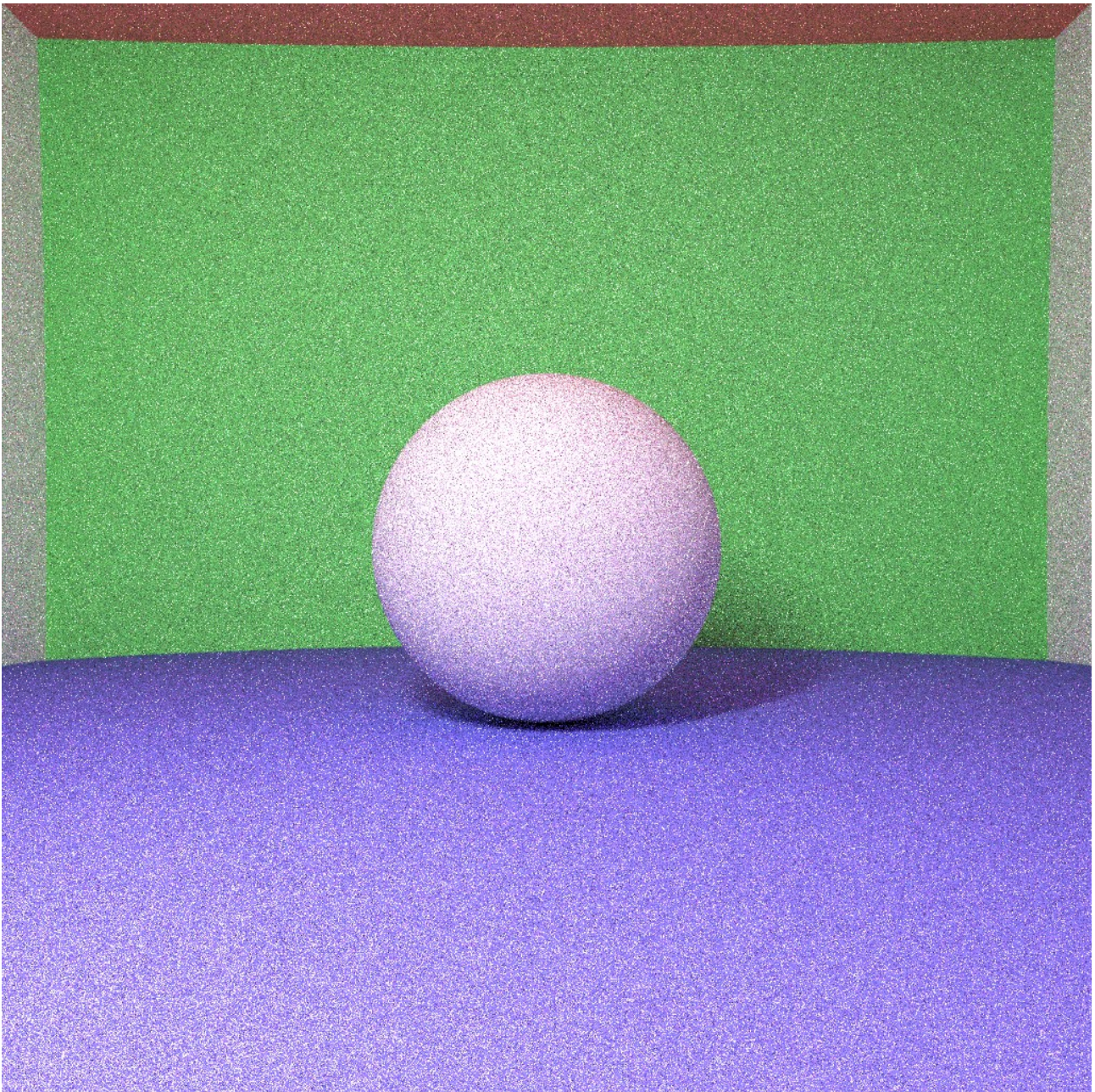




Pour avoir une image correcte en un temps raisonnable ( 257 secondes CPU), on est obligés d'avoir une source lumineuse très importante puisque la taille de la sphère joue directement sur la chance qu'un rayon lumineux touche la sphère.

On ajoute alors la contribution directe de la lumière. Le résultat est alors :





Réalisée en 185 secondes (temps CPU)

#### 4 – Différents matériaux

On peut ensuite décider de changer le matériau traité pour obtenir des surfaces spéculaires et transparentes. En effet, il suffit de connaître la BRDF de chaque surface pour adapter notre code. Ainsi, on peut voir ici des sphères diffuses, spéculaires, et transparentes.





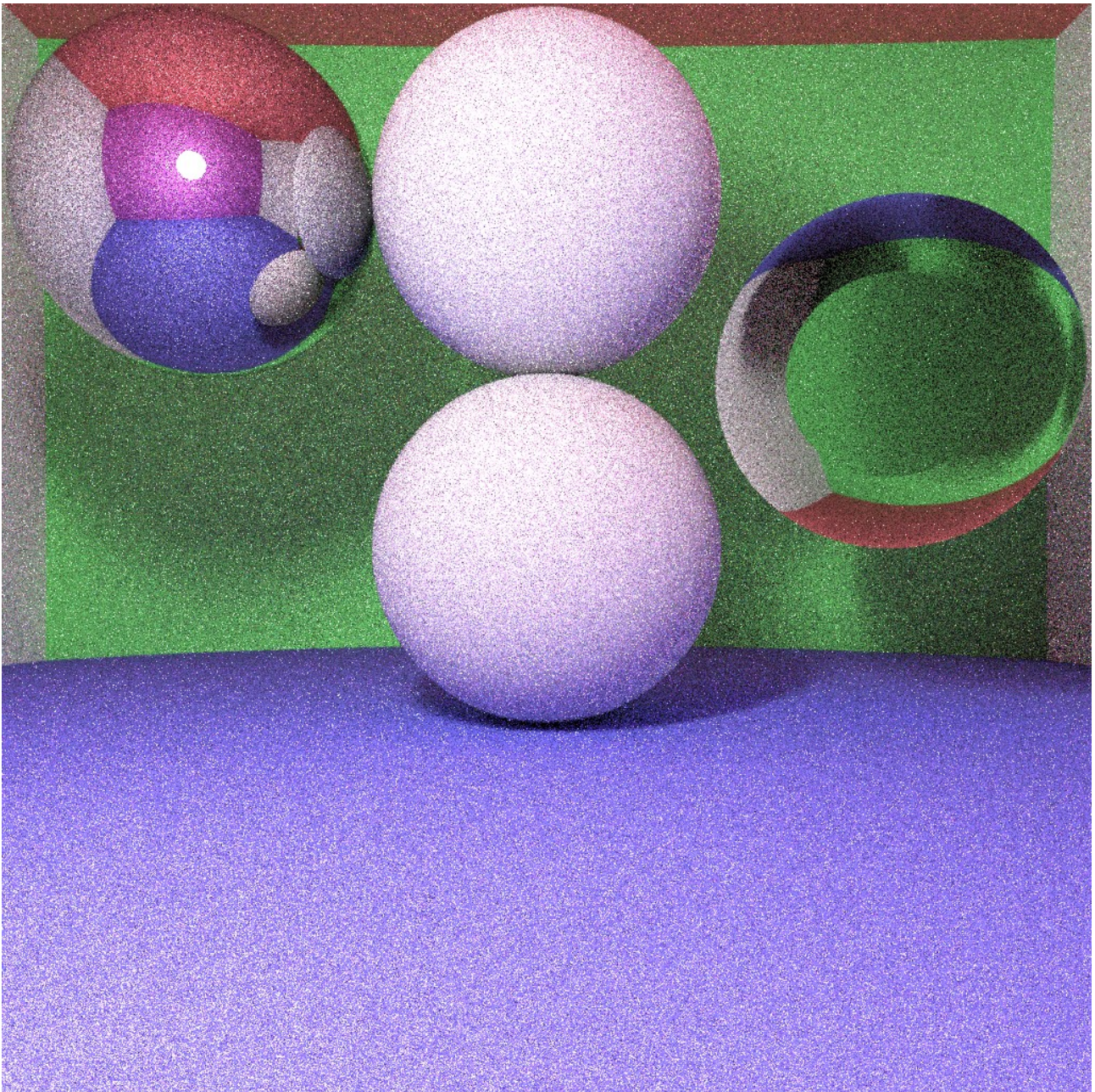


Image réalisée en 212 secondes (Temps CPU)

##### 5 – Différentes géométries

On peut également en étudiant les éléments géométrique de nature triangulaire s'intéresser aux maillages et notamment à la lecture de fichiers obj.

On peut alors à partir du fichier obj, avec les normales, un ensemble de textures et les coordonnées UV tracer des géométries complexes de façon réaliste :



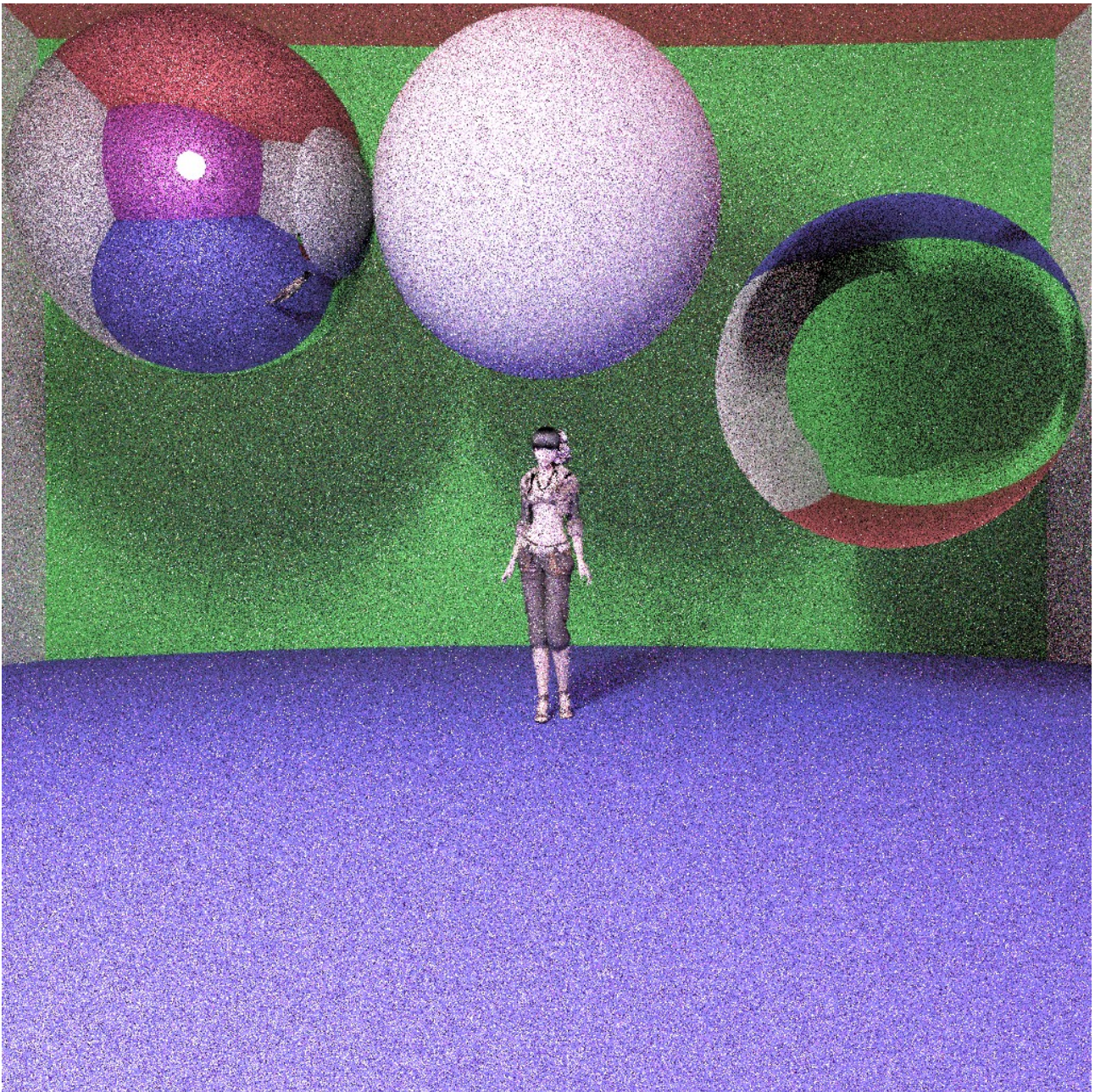


Image réalisée en 144 secondes (temps CPU)

Sachant que si nous voulions avoir des images le plus réaliste possible, il faudrait envoyer encore plus de rayons par pixel.

Par exemple, pour 100 rayons par pixel :



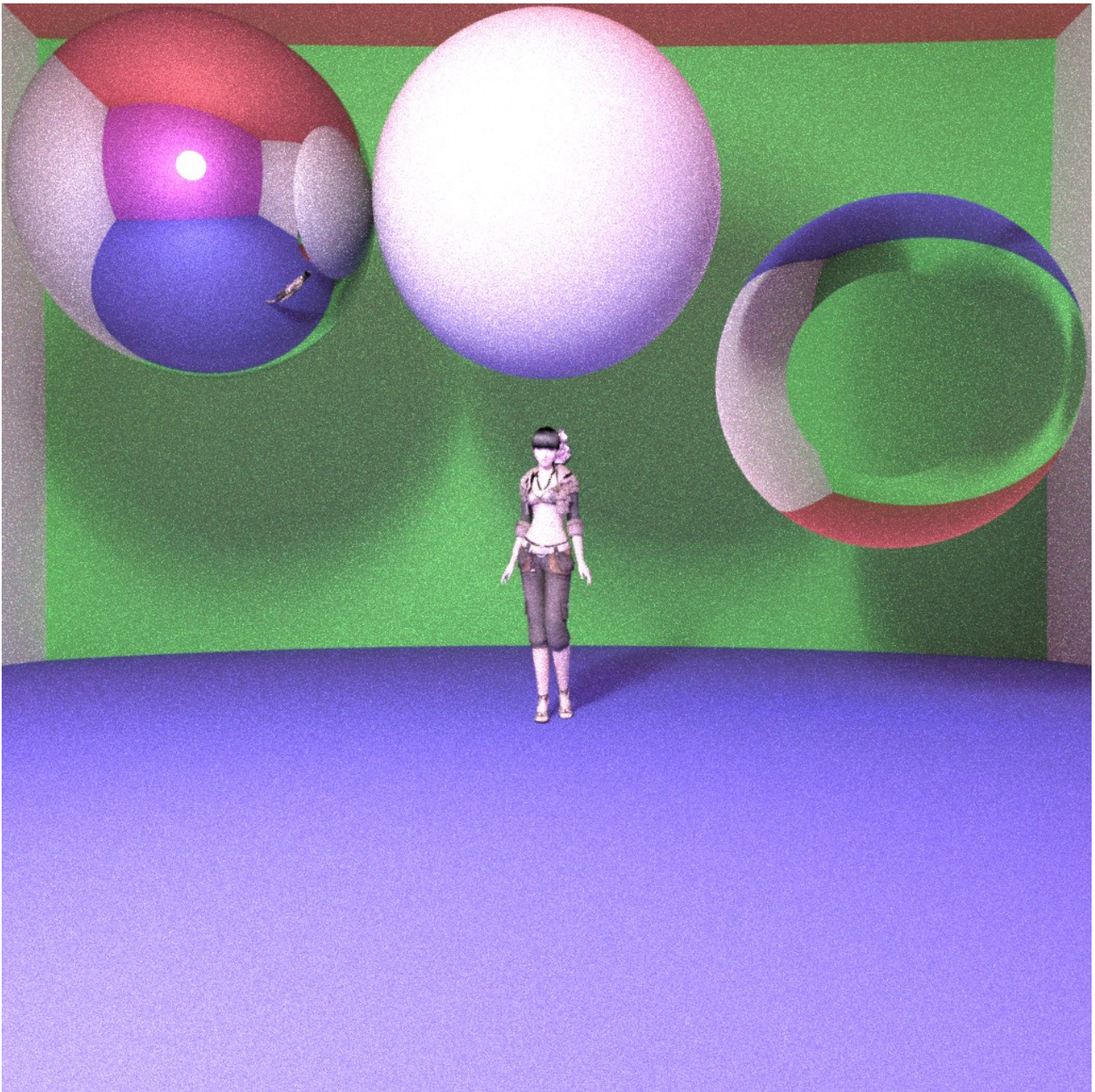


Image réalisée en 803 secondes (temps CPU)

Ceci augmente le temps de calcul considérablement, mais permet une plus grande précision dans notre résultat final.