

Raycast

Pour chaque pixel de l'écran

 Lancer un rayon

 Prendre l'intersection la plus proche de la caméra

 Calculer les coordonnées u,v inverses

1 { Pour chaque source lumineuse
 Calculer la contribution (Diff Spec)
Affichage

Raycast + Occultation

1 devient :

 Pour chaque source lumineuse

 Lancer un rayon depuis le point courante vers la source

 Si pas d'occultation

 Calculer la contribution de cette lampe (Diff Spec)

Production d'ombre dure

Ajout de la Réflexion

A la suite de 1 la contribution du diffus et la contribution du spéculaire ont été calculées. Nous avons comme couleur finale :

$$\text{Couleur finale} = \rho_{\text{diffus}} \times \text{Contrib}_{\text{diffus}} + \rho_{\text{spec}} \times \text{Contrib}_{\text{spec}}$$

Pour un objet mat, ρ_{spec} est nul, pour un objet transparent, ρ_{diffus} est nul.

La réflexion et le spéculaire correspondent physiquement à la même notion, la différence entre ces deux notions est pour nous uniquement algorithmique. En effet, ce que nous appelons réflexion correspond à la lumière réfléchie depuis un autre objet (effet miroir) et ce que nous appelons spéculaire correspond à la lumière réfléchie depuis une lampe (neon glow, carrosserie de voiture). Nous les codons séparément car les sources de lumière n'ont pas d'existence physique. La formule du spéculaire que vous avez utilisée au début correspond au spéculaire produit par les lampes de la scène, il nous reste à rajouter celui des objets :

$$\text{Couleur finale} = \rho_{\text{diffus}} \times \text{Contrib}_{\text{diffus}} + \rho_{\text{spec}} \times \text{Contrib}_{\text{spec}} + \rho_{\text{ref}} \times \text{Contrib}_{\text{ref}}$$

Après 1, ajout

 R : vecteur camera>point courant

 Calcul du symétrique S de R par rapport à N

 Lancer un rayon depuis le point courant dans la direction S

 Calcul de l'intersection la plus proche et calcul de sa couleur > $\text{Contrib}_{\text{spec}}$

Attention, un miroir n'a pas de diffus !

Ajout du pathtracing

Le modèle d'illumination diffus est très approximatif. En effet, on ne considère que la contribution direction des rayons des lampes alors que les rayons lumineux proviennent de toutes les zones de la scène. Nous faisons donc un moyennage des couleurs provenant d'un échantillonnage de rayons. Nous choisissons une distribution de rayons telle que les directions vers le pôle soient surreprésentées, ainsi en moyennant simplement la contribution de chaque rayon, on fait apparaître indirectement le facteur en $\cos \theta$ présent dans la formule simplifiée du diffus.

Une fois pour toute :

Calcul d'un ensemble de n points S régulièrement répartis sur une sphère unitaire

Remplacer le calcul du diffus par :

```
Calcul de la normale N de l'objet // il faut un vecteur UNITAIRE
CoulFin = 0
Pour chaque vecteur s dans S
    v = s + N
    Lancer un rayon de direction v depuis le point courant
    Calcul de la couleur C de l'intersection suivant les formules simplifiées
    CoulFin += C
CoulDiffus = CoulFin / n
```

Si le rayon heurte une lampe, ajouter le niveau de luminosité de la lampe. Les lampes ont maintenant des existences physiques dans la scène (rectangle, sphère). Plus une lampe est proche d'un objet, plus de rayons vont l'intersecter, plus elle va l'éclairer.

Pour le calcul de la couleur C , nous utilisons les formules traditionnelles. Ainsi, nous avons besoin de lampes ponctuelles ou directionnelles. Vous devez donc en parallèle de vos area lights créer des lampes ponctuelles ou directionnelles qui les approximent au mieux pour le calcul de luminosité simplifiée. Vous pouvez les positionner au centre des nouvelles lampes.

A noter que ce process remplace le calcul d'occultation.

Pour améliorer la qualité de l'image et éviter les effets de discrétisation de l'ensemble de points S , rajoutez un léger bruit sur les composantes du vecteur v chaque fois qu'il est généré. Un peu d'aléatoire va lisser l'image.

Prenez $n < 150$ pour avoir des temps acceptables