Projet lancé de rayon

# Introduction

1) On considère un rayon partant de A et dirigé selon u. Sa représentation paramétrique est r(t)=A+tu, t>0

2) On place une sphère de centre C et de rayon R. Calculer H, le projeté de C sur r(t)

3) Si HC>R, le rayon évite la sphère. Sinon on calcule les deux intersections (ou une, si tangence) en remplaçant les coordonnées de r(t) dans l'équation cartésienne de la sphère.

4) On garde celle qui est la plus proche de A.

5) on applique le modèle d'illumination par une source ponctuelle en ce point. La normale en ce point est évidente. Si B est le point d'intersection, on prendra comme normale unitaire le vecteur CB divisé par sa norme.

Une couleur n'est visible que si elle reçoit de la lumière. C'est pourquoi, si un pixel n'est pas dans l'ombre d'un autre objet, on détermine la source de lumière qui l'éclaire.

# Lumière ambiante :

La lumière ambiante ne permet pas d'afficher les reliefs d'un objet 3D.

Par exemple, une sphère éclairée par une lumière ambiante, sera représentée par un cercle plein en 2D sur l'image.

Il va donc falloir partir de l'équation cartésienne d'une sphère et aboutir à une équation du second degré que l'on résout avec le calcul de discriminant.

Tout d'abord, on connaît :

- La position de la caméra

- La position de la lumière

- Le vecteur d'un rayon de lumière

- L'origine de la sphère à éclairer

- Le rayon de cette dernière

- La distance focale

- L'équation cartésienne d'une sphère : (X-Xc)² + (Y-Yc)² + (Z-Zc)² = r²

- Calcul de discriminant :

Soit l'équation : aX² + bX + c = 0

On calcul le discriminant (Delta) :

Delta = b² – 4ac

Si Delta < 0 :

Aucune solution, donc pas d'intersection.

Si Delta = 0 :

Une solution, donc une intersection :

solution = -b² / 2a

Si Delta > 0 :

Deux solutions, donc deux intersections :

solution1 = (-b + Delta) / 2a

solution2 = (-b - Delta) / 2a

Une fois cette intersection mise en évidence on récupère le point le plus proche de la caméra si on a 2 point et on l’illumine en fonction de l’intensité de la lumière **Ip** et **Ka** est le coefficient ambiant de la surface (le nombre de lumière renvoyée).

On se retrouve donc avec la formule suivant : Ia = Ip \* Ka

## Remarque :

Pour l’instant toutes les sphères sont illuminés mêmes si d’autre obstacle sont entre la lumière et la sphère. On fait juste un lancer de rayon sans prendre en compte les autres éléments, il n’y a par conséquent pas d’ombre.

# Lumière Diffuse :

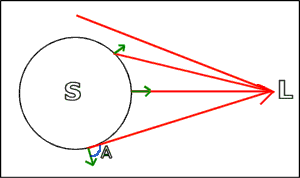
La couleur d’un objet est directement issue de la quantité de lumière qui le touche et des caractéristiques physiques de l’objet vis à vis des longueurs d’onde qu’il diffuse.

On utilise donc la partie lumière diffuse du modèle de Phong pour que la diffusion soit effectuée isotropiquement (uniformément dans toutes les directions).

Il n’y a pas plus de lumière diffusée dans une direction que dans une autre.

Deux directions d’incidence différentes peuvent conduire à la même lumière diffusée (si l'angle d'incidence est égal pour les deux directions, voir plus loin).

L’éclairage d’un point **P** d’une surface **S** dépend uniquement des caractéristiques matérielles de **S** et de l’angle **téta** (angle d'incidence) entre le vecteur normé vers la source de lumière en **P** et la normale **N** extérieure à **S** en **P**. Le vecteur **L**  quant à lui représente le vecteur **PL** qui va du point **P**  à la source lumineuse.



L’intensité est donnée par la formule **is = Ip Kd (L.N)**

**Ip** est l’intensité de lumière (énergie lumineuse par unité de surface) reçue en **P** et **Kd** est le coefficient de diffusion de la surface.

L'utilisation de **(L.N)** permet de caractériser mathématiquement le fait que plus la lumière est zénithale, plus elle éclaire fortement une surface. Au contraire, plus la lumière est rasante, moins elle éclaire cette même surface.

Au début nous avions trouvé une autre formule : **is = Ip Kd \*cos(théta)** mais pour calculer l’angle nous devions repasser par du produit scalaire. Pour des soucis d’optimisation nous avons donc choisi de garder la première formule.

## Remarque :

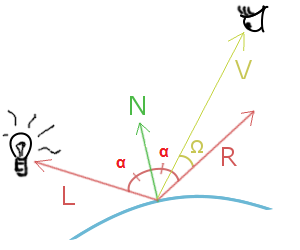
Tous les vecteurs doivent être normalisés pour faire les calculs afin de retrouver exactement le cosinus de **théta** quand on fait **(L.N).**

# Lumière spéculaire

La lumière spéculaire est la partie de la lumière qui est renvoyé pile vers la caméra, à 90° comme dans un miroir. C’est cette lumière qui va donner une réflexion sur la forme d’une tache blanche sur la sphère.

Pour calculer cette nouvelle lumière nous avons besoin de nouvelles variables. **Ks** nous le coefficient spéculaire de la surface, **alpha** qui est lui lié à la brillance de la surface. Plus **alpha** est grand plus la surface est brillante, ce n’est pas un coefficient par conséquent il peut prendre des valeurs élevées.

Enfin nous nous avons besoin du vecteur **V** pour la direction de vue de l’observateur (ici la caméra) et le vecteur **R** afin de savoir dans quel sens la lumière est réfléchie. Le dessin ci-dessous reprend ces différentes variables.



Le problème c’est que nous n’avons pas directement le vecteur R. Nous l’obtenons grâce au simple calcul suivant : **R** = 2(**N**.**L**)**N**-**L**

Enfin pour obtenir l’intensité on effectue le calcul suivant :

**Is**= **Ip** **Ks**(**R**.**V**)**alpha**

## Remarque :

On se rend vite compte que si V = R alors la tache spéculaire est pile en face de l’observateur et l’intensité est à son maximum.

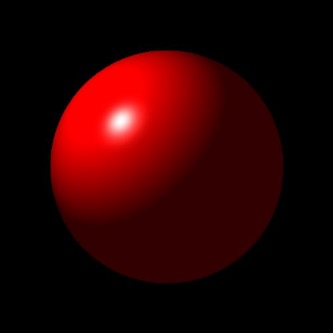
.

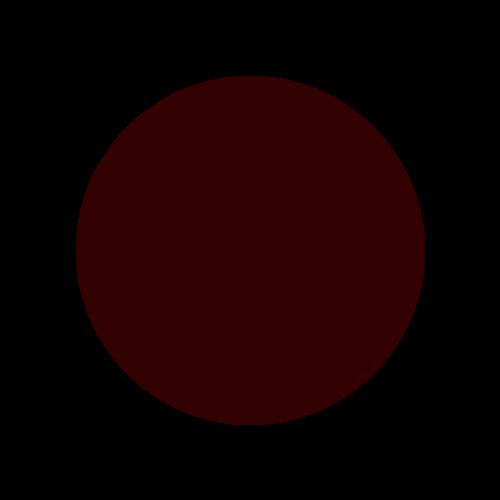
# Ombrage de Phong

Le modèle d’Ombrage de Phong et basé sur le mélange de la lumière ambiante, de la lumière diffuse et de la lumière spéculaire. Ainsi pour avoir l’intensité globale on additionne les 3 intensités différentes, on a donc :

**I** = **Ia**+**Id** + **Is**

Si dessous on peut vous les trois différentes lumières dans notre projet python et leur addition :





|  |  |  |  |
| --- | --- | --- | --- |
| **Lumière ambiante** | **Lumière diffuse** | **Lumière spéculaire** | **Illumination de Phong** |

# Ombrage

Pour avoir un rendu plus réaliste, surtout au niveau des sphères que cache la lumière à d’autres, nous avons choisi d’implémenter tout la partie ombrage.

Le calcul est assez simple, comme précédemment on parcourt tous les pixels de notre camera, on lance un rayon, on trouve s’il y a intersection ou non avec une sphère et on récupère celle la plus proche.

La partie qui diffère est que quand on récupère de vecteur **L,** on vérifie si celui-ci croise une autre sphère et on récupère le point d’intersection **P’** le plus proche.

Ensuite on compare la norme de **PP’** avec la norme du vecteur **L**. Si la norme de **PP’** est plus petite alors le point de la première sphère n’est finalement pas illuminé.

Mais pour que cela marche complètement il faut vérifier que (**N.PP’)** est bien supérieur à zéro. En effet même si une autre sphère est plus proche de la sphère que la lumière rien ne nous dit qu’elle est entre la sphère et la lumière est non derrière. En calculant ce scalaire nous nous en assurons.

# Fonctionnement du programme

Notre programme en Python reprend les calculs expliqués au-dessus. Nous avons suivis le sujet et avons créé les classes suivantes :

* Point
* Vector
* Color
* Ray
* Camera
* Light
* Sphere
* Scene

C’est dans la classe scène que se passe le plus gros du travail. Celle-ci contient tous les éléments de la scène, les lumières, les sphères la caméra. Nous avons défini une méthode « process » qui parcourt toutes les sphères et les lumières et qui illumine chaque pixel de la camera.

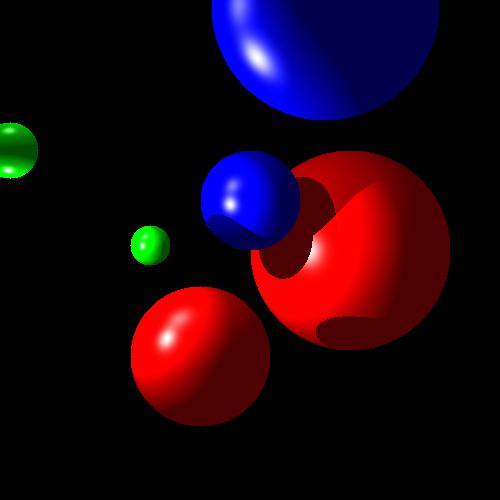
A la fin nous récupérons donc une matrice avec les couleurs de tous les pixels de notre image.

# Répartition du travail

Nous avons surtout travaillé en groupe, le midi ou entre 2 cours. Ainsi il est compliqué de dire qui exactement à fait quoi étant donné que nous avons toujours travaillé en groupe.

# Rendu Final

Ci-dessous nous vous mettons un exemple de ce que nous avons finalement fait avec l’implémentation Python. La scène est composée de différentes boules de différentes couleurs et de 2 lumières de Phong. Une lumière de Phong reprend à la fois la lumière ambiante, la lumière diffuse et la lumière spéculaire, elle fait les 2 d’un coup :



Une seconde avec une intensité de 1

Une première lumière avec une intensité de 0.5