

Lancé de rayon

Le but de ce projet est de faire un lancé de rayon simplifié :

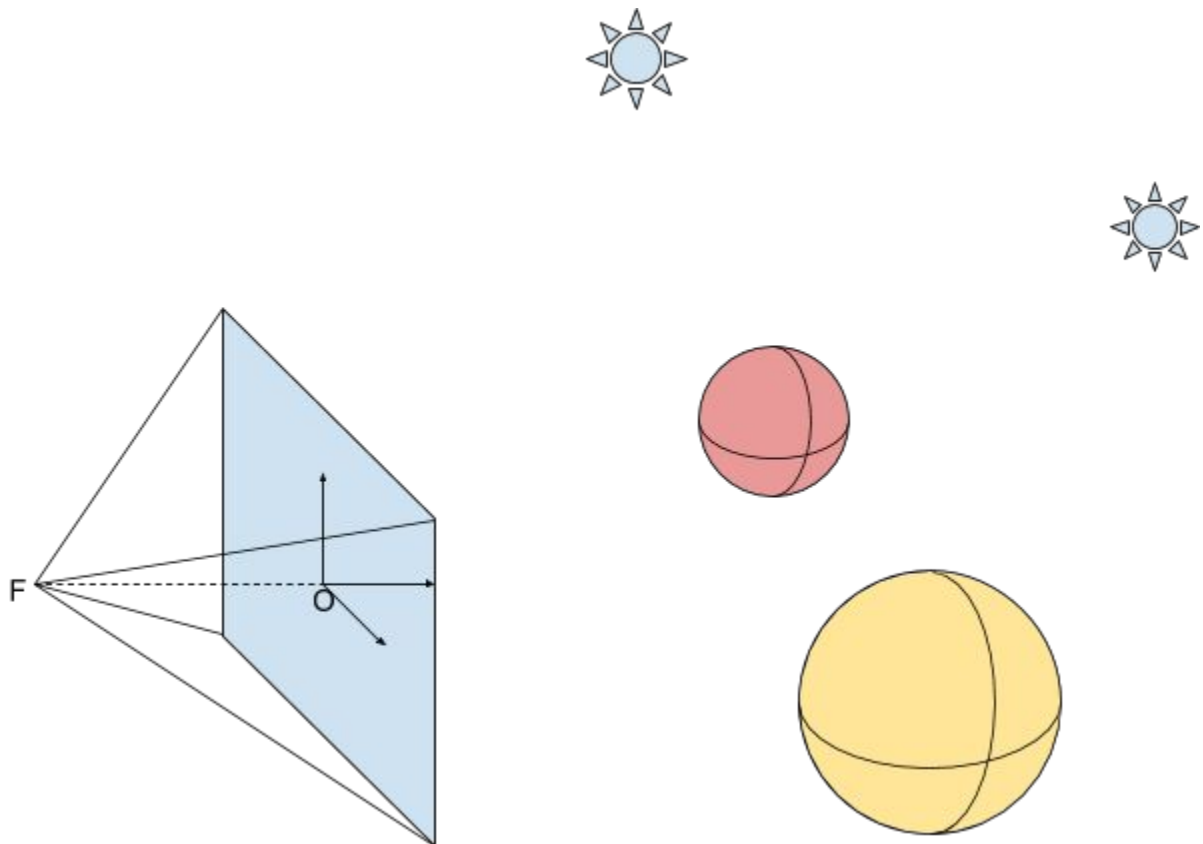
- les seuls objets de la scène sont des sphères
- les seules lumières sont ponctuelles
- tout se passe dans le référentiel de la caméra

Le principe est simple : notre caméra est un écran composé de pixels.

Pour chaque pixel :

1. on génère un rayon, qui part du centre focal et qui passe par ce pixel
2. on calcule si des sphères intersectent le rayon, et si c'est le cas, quel est le point d'intersection le plus proche de l'écran.
3. a partir de ce point d'intersection, des positions des lumières, et de la couleur de la sphère en ce point, on calcule la couleur du pixel

Sur le schéma ci-dessous, on représente une scène :



On observe que la caméra est toujours centrée en O, l'origine du référentiel global, et orientée selon le vecteur y.

Le point F est le centre focal : il est de coordonnées $(0, -f, 0)$, f étant la distance focale.

Tips : Pour bien comprendre la position spatiale d'un pixel, réfléchissez à la position d'un pixel dans un écran, en fonction de la taille et de la position de celui-ci et de sa résolution.

Vous devez créer des classes:

- Point : un point 3d avec donc trois coordonnées X,Y,Z et toutes les méthodes qui vous semblent nécessaires
- Vector : un vecteur 3d donc trois composantes X,Y,Z et toutes les méthodes qui vous semblent nécessaires
- Color : une couleur qui contient trois composantes R, G, B
- Ray : un rayon composé d'un Point et d'un Vector
- Camera : une caméra, qui comporte une résolution (screen_width et screen_height), une largeur et une hauteur (width et height) et une distance focale.
- Light : Une lumière, composée d'un Point et d'une Color
- Sphere : une sphère, composée d'un Point et d'un rayon, et d'une Color, et de coefficients nécessaires aux formules d'illumination
- Scene : contient une Camera, un tableau de Light et un tableau de Sphere. On doit pouvoir ajouter de nouvelles Light et Sphere (en utilisant l'opérateur +, ou une méthode si vous n'y arrivez pas). Doit contenir une méthode qui génère l'image de la scène et la retourner (tableau de pixels)

Concernant le modèle d'illumination, essayez de le faire de façon incrémentale, en rajoutant des parties au fur et à mesure :

- lumière ambiante
- lumière diffuse
- lumière spéculaire
- ombres
- réflexions
- transparence

Vous ferez un main qui permet de tester vos classes, en créant une scène "à la main" et en sauvegardant le résultat dans un JPG (utilisez la librairie Pillow).

En bonus, vous pouvez faire un loader de scène qui parse un json ou un xml décrivant la scène.

En plus des sources, vous rendrez un rapport contenant :

- les paramètres potentiels à mettre en ligne de commande
- les explications du fonctionnement de votre programme,
- son architecture
- qui a fait quoi,
- les équations utilisées