



Travaux dirigés d'infographie n°5

Cours de synthèse d'image I

— IMAC première année —

Lancer de rayon 1

Durée : 4 heures

Dans cette séance nous aborderons les premières étapes du lancer de rayon.

► Exercice 1. Construction des structures

Créez des structures pour les objets / concepts suivants :

- Un point 3D
- Un vecteur 3D
- Un rayon
- Une couleur
- Une sphère
- Un cône (infini)
- Un plan
- Un cylindre

La sphère, le cube, le plan, le cylindre et le cône seront des objets de la scène. Un objet sera représenté par la structure suivante :

```
typedef union tgen_objet {  
    int type;           // Indique le type de l'objet;  
    tcone cone;         // Mettre  
    tsphere sphere;     // vos propres  
    tplan plan;         // noms  
    tcyl cyl;           // de type  
} objet;
```

La structure d'union permet de regrouper en un même type plusieurs structures ayant un "concept" commun (ici celui d'objet de la scène). La place mémoire réservée par le compilateur pour une union est celle du plus grand élément de l'union. Vous utiliserez une série de `#define` pour encoder les différents types d'objet comme

```
#define T_CONE 1
...
```

► Exercice 2. Calcul d'intersection

Les fonctions d'intersection prendront en argument un rayon et un objet et retourneront vrai si l'objet est touché par le rayon et faux sinon. De plus, les fonctions devront calculer et retourner le point d'intersection. Vous réaliserez au moins les 3 fonctions d'intersection sur le plan, la sphère et le cône.

► Exercice 3. Création de la scène

Définissez une scène contenant :

- Le plan $y = -1$
- Une sphère de rayon 0.7 et de centre 0.5 -0.5 0
- Un cylindre de sommet -0.5 1 -1, d'axe 0 -1 0 de rayon 1 et de longueur 2

Puis, associez une couleur à chacun de ces objets.

► Exercice 4. Création de l'image

Nous allons décomposer le problème en sous tâches :

- D'abord, faites une boucle qui calcule pour chaque position d'un pixel à l'écran le rayon correspondant sachant que la caméra est placée en 0 0 2, qu'elle regarde dans la direction des Z négatifs, que l'écran est situé à 0.1 du centre de la caméra, qu'il a une taille de 0.1 sur 0.05 et qu'il contient 256x128 pixels.
- Puis à l'aide de cette boucle et de vos fonction d'intersection, testez si chaque rayon intersecte une surface.
- Maintenant, sur tous les objets interceptés par le rayon, gardez, pour chaque pixel, la référence de la couleur de l'objet le plus proche de la caméra. Si le rayon n'intersecte pas d'objet, la couleur sera noire (ou une autre couleur symbolisant le fond de la scène).
- Enfin, à l'aide de toutes les couleurs obtenues, faire une image (en .ppm) de la scène.



► Exercice 5. Introduction aux coordonnées homogènes

Demandez l'explication sur les coordonnées homogènes. Puis créez une structure de matrice 4x4. Enfin, réalisez 4 fonctions :

- `creerMatTranslation` qui prend en argument un vecteur et renvoie une matrice 4x4 de translation.
- `creerMatRotX` qui prend en argument un angle et renvoie une matrice de rotation autour de l'axe X.
- `creerMatRotY` qui prend en argument un angle et renvoie une matrice de rotation autour de l'axe Y.
- `creerMatRotZ` qui prend en argument un angle et renvoie une matrice de rotation autour de l'axe Z.

Puis réalisez les fonctions suivantes :

- Multiplication d'un point par une matrice 4x4.
- Multiplication d'un vecteur par une matrice 4x4.
- Transformation d'un rayon par une matrice 4x4.
- Multiplication de deux matrices 4x4.

Maintenant nous allons définir une scène contenant les 3 objets suivants :

- Une sphère de centre 0 0 0 et de rayon 0.5 translattée de -0.5 0 0.
- Un cylindre de sommet 000 d'axe 001, de rayon 1.0 et de hauteur 1.0. Le cylindre sera sera pivoté de 45 degre par rapport à l'axe des X.
- Un plan $y = 0$ subissant une translation de -3 0 0 et une rotation sur l'axe des Z de -10 degrés
- À quoi peuvent servir ces matrices ?
- Où doit-on intégrer cette utilisation de matrice ?
- Comment stocker les matrices ?

Enfin réalisez l'algorithme de calcul de cette nouvelle scène en utilisant les matrices 4x4.