COMPILATION D'OBJETS 3D

Sylvain HUET

janvier 1994

1 Présentation du problème.

La représentation tridimensionnelle d'objets en temps réel nécessite ou bien un matériel spécialisé, ou bien une grande puissance de calcul. Le Z-Buffer requiert une mémoire importante et est de ce fait mal approprié aux hautes résolutions sur des systèmes modestes. Le Ray-tracing ne permet pas encore d'obtenir du temps réel. Reste le traditionnel algorithme du peintre, dont la phase de tri des surfaces, en n^2, limite rapidement la complexité des scènes. Cette phase de tri peut cependant être fortement réduite par une phase initiale de compilation : c'est ce que la méthode présentée ici se propose de réussir.

2 Préliminaires.

2.1 Domaine d'utilisation.

Les objets que nous considérerons seront simplement constitués de triangles dans l'espace. Nous supposerons encore ces objets totalement rigides et sans articulation. La méthode prendra en entrée la liste des triangles et la liste des sommets.

2.2 Définitions géométriques.

Définition 0 : triangle

On appellera "triangle" la partie d'un hyperplan (que nous appellerons le support du triangle) délimitée par trois segments. Un triangle sera donc caractérisé par trois points. "L'intérieur" du triangle fait donc partie de celui-ci.

Définition 0 bis : fonction sgn()

La fonction sgn() définie sur l'ensemble des réels associe à x :

- 1 si $x > 0$

- 0 si $x = 0$
- -1 si $x < 0$

Définition 1 : les "côtés" d'un triangle.

Le vecteur normal n d'un triangle T, de sommets A, B et C (colinéaire au produit vectoriel de AB par AC), définit les "côtés" du triangle T. Soit un point O :

- O est du côté + de T si sgn(AO.n)$\geq$0
- O est du côté - de T si sgn(AO.n)$\leq$0
- (O peut donc être des deux côtés de T)

Soient deux points P et Q, P et Q sont du même côté de T si l'une des trois conditions suivantes est remplie :

- sgn(AP.n)=sgn(AQ.n)$\neq$0
- sgn(AP.n)=0
- sgn(AQ.n)=0

En particulier, la position O de l'observateur, définit pour chaque triangle le côté de l'observateur.

Soient deux triangles U et V : U et V sont du même côté de T si les sommets de U et V sont tous du même côté de T.

Définition 2 : ensemble convexe de triangles.

Un ensemble convexe de triangles est un ensemble E de triangles, tel que pour tout triangle T de E, les triangles de E sont tous du même côté de T.
Remarque : les triangles de E sont sur l'enveloppe convexe de E. Dans la suite, on supposera que le vecteur normal des triangles de E est dirigé vers l'intérieur de l'ensemble. On parlera alors de l'intérieur et de l'extérieur d'un triangle T.

2.3 Propositions.

Proposition 1 : tri de deux points d'un ensemble convexe.

Soient A_1 (resp.A_2) un point d'un triangle T_1 (resp.T_2), d'un ensemble convexe E.

Supposons que l'observateur O se trouve sur la droite (A_1, A_2). Essayons de déterminer quel point, vu de O, se trouve devant l'autre. Il y a quatre possibilités :

- O se trouve à l'intérieur de T_1 et de T_2, alors :

  - ou bien $A_1 = A_2$, et l'ordre importe peu.
  - ou bien O ne peut voir les deux points à la fois : O se trouve entre A_1 et A_2.

- O se trouve à l'intérieur de T_1 et à l'extérieur de T_2, alors :
  - A_1 se trouve derrière A_2.
- O se trouve à l'extérieur de T_1 et à l'intérieur de T_2, alors :
  - A_2 se trouve derrière A_1.
- O se trouve a l'extérieur de T_1 et de T_2, alors :
  - $A_1 = A_2$ et l'ordre importe peu.

On en déduit que l'affichage d'un ensemble convexe de triangles peut se faire dans l'ordre qui suit :
1- Afficher dans un ordre quelconque les triangles à l'intérieur desquels l'observateur se trouve.
2- Afficher dans un ordre quelconque les triangles à l'extérieur desquels l'observateur se trouve.

Cet ordre reste correct même lorsque l'observateur se trouve à l'intérieur de l'ensemble convexe.

Proposition 2 : tri d'un point intérieur à un ensemble convexe.

Soit M un point intérieur à E (strictement) : où que soit l'observateur, le point M se trouve devant les triangles de E dont il voit l'intérieur, et derrière ceux dont il voit l'extérieur.
On en déduit un nouvel ordre d'affichage :
1- Afficher dans un ordre quelconque les triangles de E à l'intérieur desquels l'observateur se trouve.
2- Afficher -dans un ordre à déterminer- les triangles se trouvant à l'intérieur de E.
3- Afficher dans un ordre quelconque les triangles de E à l'extérieur desquels l'observateur se trouve.

Cet ordre permet d'afficher sans "réfléchir" un objet composé d'ensembles convexes inclus l'un dans l'autre : $E_0, E_1, ..., E_n$ où $E_i + 1$ est inclus dans E_i ; la procédure appelée pour afficher l'ensemble Ei se rappelle récursivement à l'étape 2 pour l'ensemble $E_i + 1$. Le premier appel se fait avec l'ensemble E_0, et la récursion s'arrête à E_n. Un exemple intéressant d'un tel objet est l'escalier.

Proposition 3 : principe universel de tri.

Soient deux points distincts A et B, et O l'observateur. Soit un plan P séparant l'espace en deux demi-espaces, l'un contenant A, l'autre contenant B. Si O se trouve dans le demi-espace de A, alors O ne peut pas voir B devant A, et inversement.

Soit un objet composé de deux parties S_1 et S_2 séparées par un plan P, alors on applique l'ordre d'affichage suivant : - si O est du même côté de P que S_1, afficher S_2 puis S_1. - sinon, afficher S_1 puis S_2.

3 Technique utilisée.

3.1 Structure produite par la compilation.

Des propositions précédentes il apparat nettement une structure d'affichage récursif, organisée autour d'un arbre d'affichage. Deux types de noeuds :

Noeud 1 :

- ensemble convexe E de triangles.
- noeud "fils" N.

Ce type de noeud s'affiche de la manière suivante :

- Afficher dans un ordre quelconque les triangles de E à l'intérieur desquels l'observateur se trouve.
- Afficher le noeud N
- Afficher dans un ordre quelconque les triangles de E à l'extérieur desquels l'observateur se trouve.

Noeud 2 :

- plan séparateur P, orienté par son vecteur normal n.
- noeuds fils, N_1 et N_2.

Ce type de noeud s'affiche de la manière suivante :

- si O est du côté + de P afficher N_2 puis N_1.
- sinon, afficher N_1 puis N_2.

Remarque fondamentale.

Déterminer si l'observateur se trouve à l'intérieur ou à l'extérieur d'une surface nécessite un produit scalaire (a priori déjà réalisé lors du calcul des projections des triangles). Déterminer de quel côté du plan séparateur l'observateur se trouve nécessite également un produit scalaire et une addition (équation du plan de la forme $ax + by + cz + d = 0$).

Définition de la complexité.

Nous appellerons complexité d'un arbre d'affichage le nombre de noeuds de type 2 contenus dans l'arbre. Nous appelerons complexité d'un objet la complexité minimale d'un arbre permettant de l'afficher.

Ainsi, tout objet convexe est de complexité nulle : il peut s'afficher avec un arbre constitué d'un seul noeud de type 1. De même, l'escalier est de complexité nulle : il peut s'afficher avec un arbre constitué de n noeuds de type 1 où n est le nombre de marches.

3.2 Compilation.

La construction de l'arbre d'affichage se fait récursivement : la routine est appelée avec la liste L des triangles de la scène.

noeud compilation(liste de triangles L)

- Si L est vide, retourner un pointeur nul.
- Diviser la liste L en deux :
  - liste E des triangles de L appartenant à l'enveloppe convexe de L.
  - liste F des autres triangles.
- Si la liste E n'est pas vide, alors E forme un ensemble convexe de triangles. Retourner un noeud de type 1, avec E comme ensemble convexe de triangle (en prenant garde à la définition de l'intérieur des triangles : inverser éventuellement n), et comme noeud "fils" N le résultat de la compilation de la liste F (appel récursif).
- Si la liste E est vide, choisir un plan séparateur P, et créer deux listes A_1 et A_2 des triangles se trouvant du côté + (pour A_1) et du côté - (pour A_2) (éventuellement en coupant en deux certains triangles, et donc en ajoutant des points à la scène). Retourner un noeud de type 2, avec P comme plan séparateur, et comme noeuds N_1 et N_2, le résulat de la compilation des listes A_1 (pour N_1) et A_2 (pour N_2) (appels récursifs).

Remarques.

.Couper un triangle par un plan aboutit dans le cas général à un triangle et un quadrilatère, le programme ne fonctionnant qu'avec des triangles, il faut couper le quadrilatère en deux. On obtient donc en fait trois triangles à partir d'un seul.
.Le choix du plan P peut se faire de plusieurs manières : on peut par exemple prendre le triangle de F coupant le moins possible de triangles.

3.3 Terminaison.

Remarquons que le nombre de triangles à chaque étape de la récursion est susceptible d'augmenter puisque dans le cas du noeud 2, couper un triangle par le plan P fournit un triangle à la liste A_1 et deux à la liste A_2 (ou inversement). Introduisons donc une nouvelle notion.

Soit L une liste de triangles, on note %L le nombre de plans supports d'au moins un triangle de L. Montrons qu'au cours de la récursion de la procédure de compilation, la valeur %L -correspondant à la liste L passée en paramètre à la procédure- diminue strictement. Ce nombre étant entier et positif, celà suffit à démontrer la terminaison. Car %L=0 équivaut à "L est vide".

Création d'un noeud N de type 1 :

L'ensemble E n'étant pas vide, il y a au moins un triangle T dans E. Soit P son plan support. T est sur l'enveloppe convexe de L, alors tous les triangles de L et de support P sont également sur l'enveloppe convexe de L et donc dans E. Aucun triangle de F n'a donc P pour support : %F≤%L-1.

Création d'un noeud N de type 2 :

On suppose que le choix du plan P est celui décrit à la fin du paragraphe précédent. Le triangle ayant servi à choisir P ainsi que tous ceux ayant P comme support, se retrouvent dans A_1 ou A_2, par exemple A_1. Alors ils appartiennent à l'enveloppe convexe de A_1 : le fils N_1 de N est donc de type 1. Le petit-fils N_1' est appelé avec une liste K ne contenant plus de triangles de support P (puisque ceux-ci sont dans la liste E de N_1) : % K≤%L-1. De même la liste A_2 ne contient plus de triangles de support P : $\%A_2 \leq$%L-1.

CQFD.