

MÉTHODE DE LA LISTE DE PRIORITÉ

(Ou l'algorithme du peintre)

Hypothèses

- un observateur à l'infini dans la direction des z positifs
- les objets de la scène modélisés par des polyèdres non nécessairement convexes
- seules les faces avant des objets sont retenues

Principe:

- afficher, si possible, les faces avant des objets dans l'ordre des distances décroissantes par rapport à l'observateur (procéder par couches comme le peintre)
- donc définir un ordre sur les faces

RELATION D'OBSCURCISSEMENT

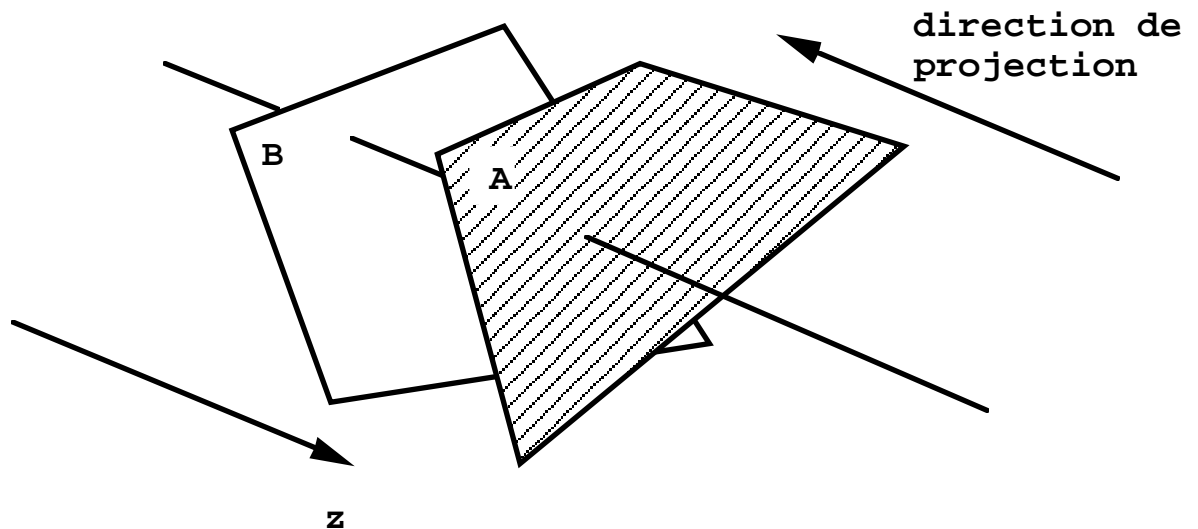
Un polygone A *obscurcit* un polygone B

$$B \ll A$$

s'il existe une droite parallèle à la direction de projection intersectant A en un point intérieur $I(A)$ et B en $I(B)$ tels que $I(A)$ se trouve plus proche de l'œil que $I(B)$.

Remarque: si A et B sont convexes, il n'est pas possible que l'on ait en même temps

$$B \ll A \text{ et } A \ll B$$



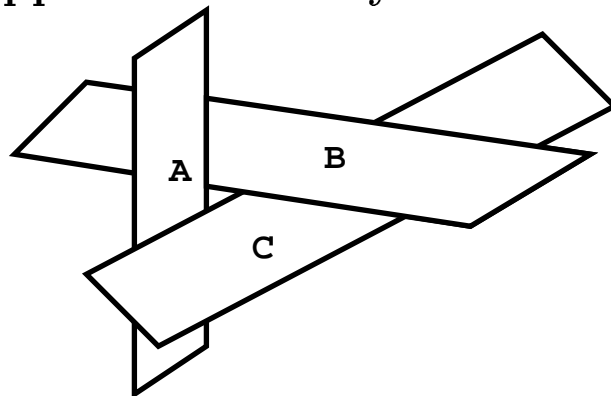
ORDRE SUR LES FACES

Prolème: définir, si c'est possible, un ordre total $<$ *compatible* avec la relation d'obscurcissement:

si $B \ll A$ alors $B < A$

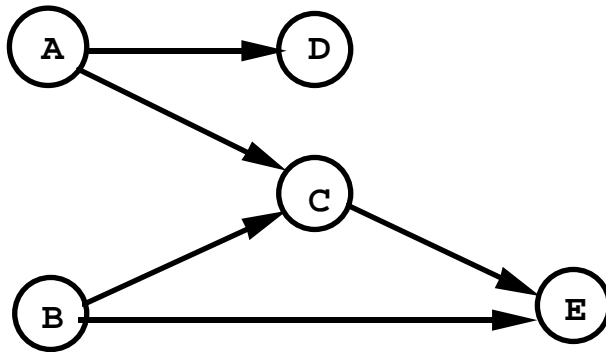
Exemple :

Apparition d'un cycle $A \ll B \ll C \ll A$



LE GRAPHE D'OBSCURCISSEMENT

- ses nœuds: les faces avant de la scène
- ses arêtes: une arête de la face A vers la face B si B obscurcit A (“l’extrémité obscurcit l’origine”)



Ordre total $<$ associé (voir tri topologique)

$A < B < D < C < E$ ou $A < B < C < E < D, \dots$

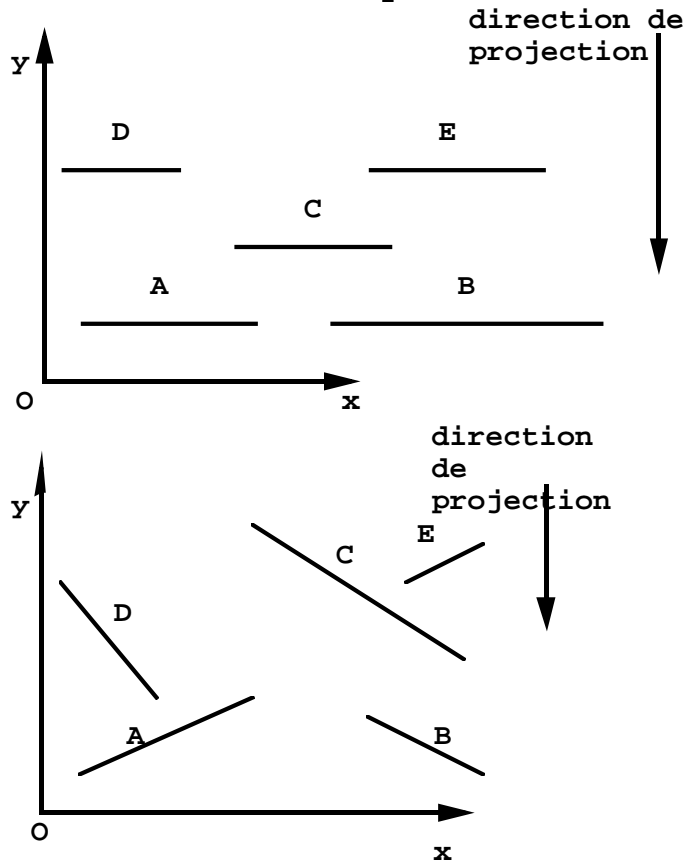
Donc on affiche, par exemple, dans l'ordre

A, puis B, puis D, puis C, puis E

ou A, puis B, puis C, puis E, puis D

ILLUSTRATON

Différentes scènes possibles associées à un graphe



UNE RELATION PLUS SIMPLE À TESTER

Remplacer la relation “B obscurcit A”,

$$A \ll B$$

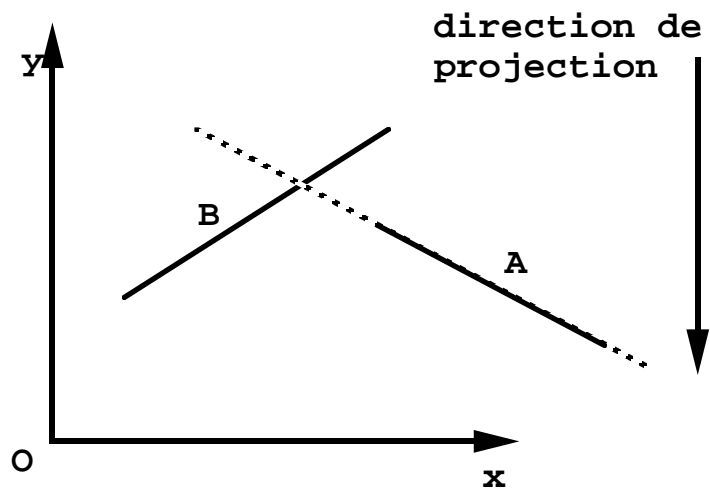
par la relation plus grossière mais plus facile à tester “B est susceptible d’obscurcir A”, $A \lll B$ si *aucune* des conditions suivantes n’est satisfaite

1. les rectangles englobant de A et B sont disjoints en x .
2. les rectangles englobant de A et B sont disjoints en y .
3. la face B est entièrement du côté opposé à celui de l’observateur par rapport au plan de A.
4. la face A est entièrement du côté de l’observateur par rapport au plan de B.
5. les projections des faces A et B sur le plan de projection sont disjointes.

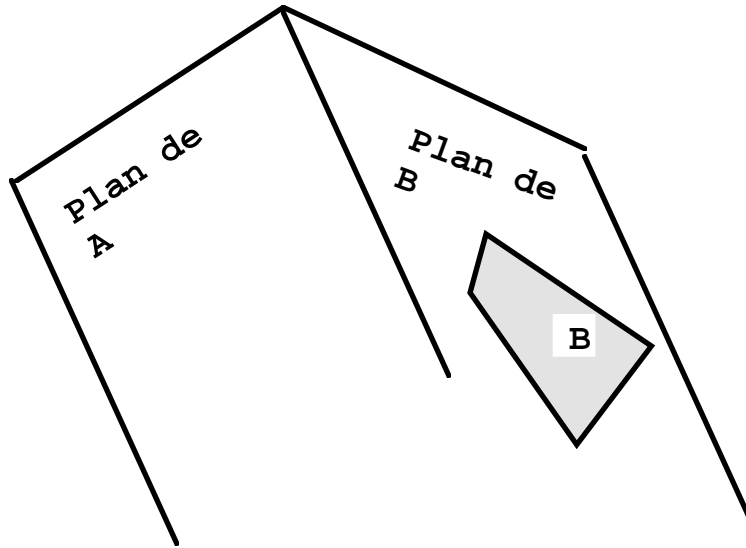
REMARQUES

- si B obscurcit A, alors B est susceptible d'obscurcir A ($A \ll B \Rightarrow A \lll B$) car aucune condition n'est satisfaite.
- on procède par coût croissant: les conditions 1, 2 et 5 sont redondantes, mais il est préférable d'éliminer 1 et 2 avant de se lancer éventuellement dans 5
- dans le plan les deux relations \ll et \lll coïncident

- les conditions 1, 2 et 5 sont symétriques mais la disjonction des conditions 3 et 4 ne l'est pas



UNE PROPRIÉTÉ IMPORTANTE



Si $A \lll B$ et $B \lll A$ sont satisfaits alors

ou bien le plan de B coupe le polygone A en plusieurs polygones A_1, A_2, \dots

ou bien le plan de A coupe le polygone B en plusieurs polygones B_1, B_2, \dots

MISE EN ŒUVRE

(Newell-Newell-Sancha) la queue Q contient les faces avant
(a priori, les faces les plus éloignées en tête)

tantque Q **non** vide **faire**

extraire le premier élément A de Q ;

(1) **pour** tout B dans Q **faire**

(2) **si** $B \lll A$ **alors**

(3) **si** B **non** marqué par A **alors début**

marquer A par B ;

échanger A et B dans Q ;

aller à sortie;

fin

(4) **sinon début** { $B \lll A$ et $A \lll B$ }

le plan de B coupe A en A_1, A_2, \dots

(resp. le plan de A coupe B en B_1, B_2, \dots)

supprimer A (resp. B) de Q ;

(5) insérer A_1, A_2, \dots (resp. B_1, B_2, \dots) dans Q ;

aller à sortie;

fin ;

effectuer la conversion de trame de A ;

supprimer A de Q ;

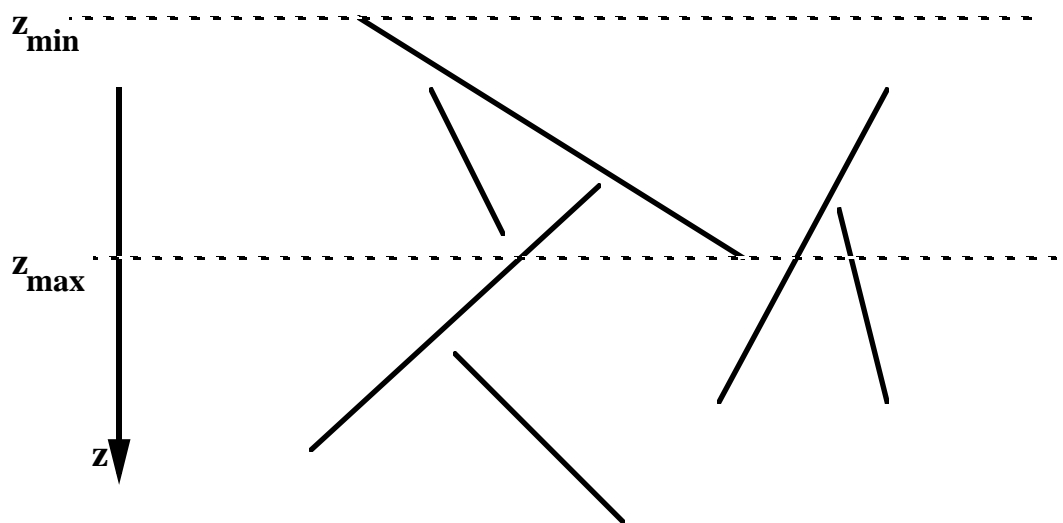
sortie: **fin**;

ARRÊT DE L'ALGORITHME

- pour une face donnée, l'instruction (3) augmente le nombre de faces qui lui sont associées
- l'instruction (4) augmente le nombre de faces découpées; or celui-ci est limité (pour chaque face initiale, considérer toutes les intersections possibles avec les plans des autres faces)
- donc l'instruction (2) est exécutée un nombre limité de fois

AMÉLIORATION DES PERFORMANCES

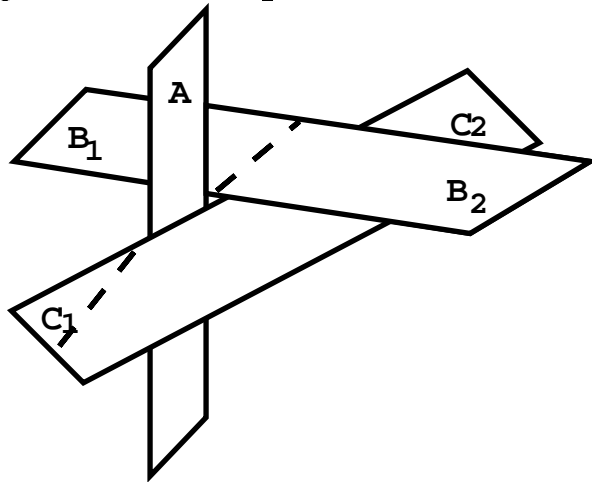
- à l'initialisation, ordonner les faces dans la queue Q par z_{\min} croissants
- dans l'instruction (1) ne considérer que les faces B dont l'intervalle en z intersectent l'intervalle en z de A



- dans l'instruction (5) insérer les nouvelles faces par z_{\min} décroissants

UNE ILLUSTRATION

Attention: certaines affirmations ne se déduisent pas de la figure, car elles font intervenir la profondeur (d'autres affirmations pourraient être compatibles avec la figure)



Etat initial de la queue

A, B, C

A est susceptible d'obscurcir B (A n'est pas derrière, B n'est pas devant):

échanger A et B et marquer A par B

$B, A(B), C$

B est susceptible d'obscurcir A (B n'est pas derrière, A n'est pas devant):

A étant marqué par B, on intersecte B par le plan de A, résultant en 2 polygones B_1 et B_2 que l'on insère dans la queue

B_1, A, C, B_2

B_1 n'est susceptible d'obscurcir aucune des 3 faces (derrière A, C, B_2):

retirer B_1 et l'afficher

A, C, B_2

A est susceptible d'obscurcir C (A n'est pas derrière, C n'est pas devant):

échanger A et C et marquer A par C

$C, A(C), B_2$

C est susceptible d'obscurcir A (C n'est pas derrière, A n'est pas devant):

A étant marqué par C, on intersecte C par le plan de A, résultant en 2 polygones C_1 et C_2 que l'on insère dans la queue

C_1, A, B_2, C_2

C_1 n'est susceptible d'obscurcir aucune des 3 faces (derrière A, B_2 , C_2):

retirer C_1 et l'afficher

A, B_2, C_2

A n'est susceptible d'obscurcir des 2 faces (derrière B_2 et C_2)

retirer A et l'afficher

B_2, C_2

B_2 est susceptible d'obscurcir C_2 (B_2 n'est pas derrière, C_2 n'est pas devant)

échanger B_2 et C_2 et marquer B_2 par C_2

$C_2, B_2(C_2)$

C_2 n'est pas susceptible d'obscurcir B_2 (C_2 est derrière)

retirer C_2 de la queue et l'afficher

$B_2(C_2)$

retirer B_2 de la queue et l'afficher