M2-Images

Transformations - Pipeline graphique

J.C. Iehl

September 29, 2010

Objectif : pipeline graphique

rappel :

- plusieurs manières d'organiser les calculs,
- mais, plusieurs parties communes.

manipuler les changements de repères, projections, ...

Objectif : géométrie

représentation d'un point, d'une direction :

- un point, noté p,
- un vecteur, noté $\vec{v}$,
- un *rayon*, noté : $r(t) = o + t \cdot \vec{d}$

représentation de l'observateur :

- une position, O et 3 directions : à droite $(\vec{X})$, en haut $(\vec{Y})$, et devant $(-\vec{Z})$,
- c'est à dire un repère orthonormé.
- le rayon et les "objets" doivent tous être représentés dans ce repère.

Rappels : points, vecteurs et repères

espace affine :

- un point : une position, pas de direction, ni de longueur,
- un vecteur : une direction et une longueur mais pas de position.

connaissant une origine, on peut construire un vecteur représentant un point.

$\vec{u} = \vec{v} \Leftrightarrow |\vec{u}| = |\vec{v}|$ et $dir(\vec{u}) = dir(\vec{v})$.

Opérations : addition, soustraction, etc.

- $p + \vec{u}$ est un point,
- $q - p$ est un vecteur, (noté $\overrightarrow{pq}$),
- si $\vec{u} = q - p \Rightarrow q = p + \vec{u}$ et $p = q - \vec{u}$,
- $p + q$ n'a pas de sens.

- $1 \cdot p = p$,
- $0 \cdot p = 0$,
- $k \cdot p$ n'est défini que si $k = 0$ ou $k = 1$?

Opérations : produits scalaire et vectoriel

produit scalaire :

- $\vec{u} \cdot \vec{v} = |\vec{u}||\vec{v}| \cos\theta$
- $\vec{u_{\parallel}} = \frac{(\vec{u}\cdot\vec{v})\cdot\vec{v}}{|\vec{u}||\vec{v}|}$
- $\vec{u_{\perp}} = \vec{u} - \vec{u_{\parallel}} = \vec{u} - \frac{(\vec{u}\cdot\vec{v})\cdot\vec{v}}{|\vec{u}||\vec{v}|}$

utile pour "projetter" un vecteur sur un autre ($\vec{u_{\parallel}}$), calculer le cosinus de l'angle entre 2 vecteurs, etc.

quel est le produit scalaire de 2 vecteurs perpendiculaires ?

Opérations : produits scalaire et vectoriel

$$\vec{u} \cdot \vec{v} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = u_x v_x + u_y v_y + u_z v_z$$

$$\text{et } |\vec{u}| = \sqrt{u_x^2 + u_y^2 + u_z^2} = \sqrt{(\vec{u} \cdot \vec{u})}$$

Opérations : produits scalaire et vectoriel

produit vectoriel :

- $|\vec{u} \times \vec{v}| = |\vec{u}||\vec{v}| \sin\theta$
- $\vec{u} \times \vec{v}$ perpendiculaire aux vecteurs $\vec{u}$ et $\vec{v}$,
- (perpendiculaire au plan défini par les 2 vecteurs)

utile pour calculer la normale d'un triangle, la surface d'un triangle, etc.

Opérations : produits scalaire et vectoriel

$$\vec{u} \times \vec{v} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$$

très utile pour construire un repère orthonormé $(O, \vec{x}, \vec{y}, \vec{z})$!

Coordonnées cartésiennes

dans un repère $(O, \vec{x}, \vec{y}, \vec{z})$:

- $\vec{u} = u_x \cdot \vec{x} + u_y \cdot \vec{y} + u_z \cdot \vec{z}$
- noté $\vec{u} = (u_x, u_y, u_z)$
- $p = O + \overrightarrow{Op} = O + p_x \cdot \vec{x} + p_y \cdot \vec{y} + p_z \cdot \vec{z}$
- noté $p = (p_x, p_y, p_z)$

avec : $u_x = \vec{u} \cdot \vec{x}$, $u_y = \vec{u} \cdot \vec{y}$, $u_z = \vec{u} \cdot \vec{z}$
et $\overrightarrow{Op} = p - O$

Repère orthonormé et produit vectoriel

dans un repère $(O, \vec{x}, \vec{y}, \vec{z})$:

$$\begin{array}{llll} \vec{x} \times \vec{y} = +\vec{z}, & \vec{x} \cdot \vec{y} = 0 \\ \vec{y} \times \vec{x} = -\vec{z}, & \vec{y} \cdot \vec{x} = 0 \\ \vec{y} \times \vec{z} = +\vec{x}, & \vec{y} \cdot \vec{z} = 0 \\ \vec{z} \times \vec{y} = -\vec{x}, & \vec{z} \cdot \vec{y} = 0 \\ \vec{z} \times \vec{x} = +\vec{y}, & \vec{z} \cdot \vec{x} = 0 \\ \vec{x} \times \vec{z} = -\vec{y}, & \vec{x} \cdot \vec{z} = 0 \end{array}$$

connaissant 2 vecteurs quelconques, comment construire un repère orthonormé ?

Changement de repères

soit un vecteur $\vec{u}$:

- coordonnées dans le repère $(O, \vec{x}, \vec{y}, \vec{z})$:
- $\vec{u} = (\vec{u} \cdot \vec{x})\vec{x} + (\vec{u} \cdot \vec{y})\vec{y} + (\vec{u} \cdot \vec{z})\vec{z}$
- et dans le repère $(Q, \vec{i}, \vec{j}, \vec{k})$?
- $\vec{u} = (\vec{u} \cdot \vec{i})\vec{i} + (\vec{u} \cdot \vec{j})\vec{j} + (\vec{u} \cdot \vec{k})\vec{k}$

Changement de repères

on connait p dans le repère $(O, \vec{x}, \vec{y}, \vec{z})$,
on veut connaître ses coordonnées dans le repère $(Q, \vec{i}, \vec{j}, \vec{k})$:

$p = O + \overrightarrow{Op}$, mais aussi $p = Q + \overrightarrow{Qp}$
donc
$p = Q + \overrightarrow{QO} + \overrightarrow{Op}$

rappel : $\overrightarrow{ab} = b - a$

Changement de repères, et avec des matrices ?

rappel : espace affine, transformation affine

- une transformation T est linéaire si :
- $T(\vec{u}+\vec{v}) = T(\vec{u}) + T(\vec{v})$,
- $T(k\vec{u}) = kT(\vec{u})$

- une transformation T est affine si :
- l'image par T d'un point est un point,
- l'image par T d'un vecteur est un vecteur.

Changement de repères, et avec des matrices ?

soit T une transformation affine et un repère $(O, \vec{x}, \vec{y}, \vec{z})$:

- $\vec{u} = (u_1, u_2, u_3) = u_1\vec{x} + u_2\vec{y} + u_3\vec{z}$
- $T(\vec{u}) = u_1 T(\vec{x}) + u_2 T(\vec{y}) + u_3 T(\vec{z})$

- $p = (p_1, p_2, p_3) = O + p_1\vec{x} + p_2\vec{y} + p_3\vec{z}$
- $T(p) = T(O) + p_1 T(\vec{x}) + p_2 T(\vec{y}) + p_3 T(\vec{z})$

Changement de repères, et avec des matrices ?

T est complètement déterminée par $T(O), T(\vec{x}), T(\vec{y}), T(\vec{z})$.

$$T(\vec{u}) = \begin{bmatrix} T(\vec{x}) & T(\vec{y}) & T(\vec{z}) & T(O) \\ & & & \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ 0 \end{bmatrix} = T \begin{bmatrix} \vec{u} \\ \\ \\ 0 \end{bmatrix}$$

$$T(p) = \begin{bmatrix} T(\vec{x}) & T(\vec{y}) & T(\vec{z}) & T(O) \\ & & & \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ 1 \end{bmatrix} = T \begin{bmatrix} p \\ \\ \\ 1 \end{bmatrix}$$

Changement de repères, et avec des matrices ?

notation homogène :

$$T^h = \begin{bmatrix} T(\vec{x}) & T(\vec{y}) & T(\vec{z}) & T(O) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\overrightarrow{u^h} = \begin{bmatrix} \vec{u} \\ w \equiv 0 \end{bmatrix}$$

$$p^h = \begin{bmatrix} p \\ w \equiv 1 \end{bmatrix}$$

rappels :

$$T(\vec{u}) = T(\overrightarrow{u^h})/u^h_w$$
$$T(p) = T(p^h)/p^h_w$$

Changement de repères, et avec des matrices ?

translation d'un point p par un vecteur $\vec{u} = (u_1, u_2, u_3, 0)$

$$T(\vec{x}) = \vec{x}$$

$$T(\vec{y}) = \vec{y}$$

$$T(\vec{z}) = \vec{z}$$

$$T(O) = O + \vec{u}$$

$$\text{donc } T = \begin{bmatrix} 1 & 0 & 0 & u_1 \\ 0 & 1 & 0 & u_2 \\ 0 & 0 & 1 & u_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Changement de repères, et avec des matrices ?

on connait p dans le repère $(O, \vec{x}, \vec{y}, \vec{z})$,
on veut connaître ses coordonnées dans le repère $(Q, \vec{i}, \vec{j}, \vec{k})$:

rappel :

$$p = O + p_x \cdot \vec{x} + p_y \cdot \vec{y} + p_z \cdot \vec{z}$$
$$p = Q + p_i \cdot \vec{i} + p_j \cdot \vec{j} + p_k \cdot \vec{k}$$

$$\begin{array}{rcl} Q & = & O + q_x \cdot \vec{x} + q_y \cdot \vec{y} + q_z \cdot \vec{z} \\ \vec{i} & = & i_x \cdot \vec{x} + i_y \cdot \vec{y} + i_z \cdot \vec{z} \\ \vec{j} & = & j_x \cdot \vec{x} + j_y \cdot \vec{y} + j_z \cdot \vec{z} \\ \vec{k} & = & k_x \cdot \vec{x} + k_y \cdot \vec{y} + k_z \cdot \vec{z} \end{array} \qquad \begin{array}{rcl} T(Q) & = & O + (q_x, q_y, q_z) \\ T(\vec{i}) & = & (i_x, i_y, i_z) \\ T(\vec{j}) & = & (j_x, j_y, j_z) \\ T(\vec{k}) & = & (k_x, k_y, k_z) \end{array}$$

quelle relation entre (p_x, p_y, p_z) et (p_i, p_j, p_k) ?

Changement de repères, et avec des matrices ?

$$
\begin{aligned}
p &= Q + p_i \cdot \vec{i} + p_j \cdot \vec{j} + p_k \cdot \vec{k} \\
&= (O + q_x \cdot \vec{x} + q_y \cdot \vec{y} + q_z \cdot \vec{z}) \\
&+ p_i \cdot (i_x \cdot \vec{x} + i_y \cdot \vec{y} + i_z \cdot \vec{z}) \\
&+ p_j \cdot (j_x \cdot \vec{x} + j_y \cdot \vec{y} + j_z \cdot \vec{z}) \\
&+ p_k \cdot (k_x \cdot \vec{x} + k_y \cdot \vec{y} + k_z \cdot \vec{z})
\end{aligned}
$$

$$
\begin{aligned}
p &= O \\
&+ (q_x + p_i i_x + p_j j_x + p_k k_x) \cdot \vec{x} \\
&+ (q_y + p_i i_y + p_j j_y + p_k k_y) \cdot \vec{y} \\
&+ (q_z + p_i i_z + p_j j_z + p_k k_z) \cdot \vec{z}
\end{aligned}
$$

Changement de repères, et avec des matrices ?

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} i_x & j_x & k_x & q_x \\ i_y & j_y & k_y & q_y \\ i_z & j_z & k_z & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_i \\ p_j \\ p_k \\ 1 \end{bmatrix}$$

ou

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} T(\vec{i}) & T(\vec{j}) & T(\vec{k}) & T(Q) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_i \\ p_j \\ p_k \\ 1 \end{bmatrix}$$

Changement de repères, et avec des matrices ?

$$\begin{bmatrix} p_i \\ p_j \\ p_k \\ 1 \end{bmatrix} = \begin{bmatrix} i_x & j_x & k_x & q_x \\ i_y & j_y & k_y & q_y \\ i_z & j_z & k_z & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

quelle relation avec $T(\vec{x})$, $T(\vec{y})$, $T(\vec{z})$ et $T(O)$?

$$\begin{bmatrix} p_i \\ p_j \\ p_k \\ 1 \end{bmatrix} = \begin{bmatrix} T(\vec{i}) & T(\vec{j}) & T(\vec{k}) & T(Q) \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

Matrices de rotation

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

rotation autour d'un axe quelconque $R(\vec{u}, \theta)$?

Rappel : Composition de transformations

$$\text{repère 1} \xrightarrow[T_{12}]{} \text{repère 2} \xrightarrow[T_{23}]{} \text{repère 3}$$

on connait p dans le repère 1, quelles sont ses coordonnées dans le repère 3 ?

$$\begin{aligned} p_2 &= T_{12} p \\ p_3 &= T_{23} p_2 \\ p_3 &= T_{23}(T_{12} p) \\ p_3 &= T_{23} T_{12} p \\ p_3 &= Tp \text{ avec } T = T_{23} T_{12} \end{aligned}$$

Applications au pipeline graphique

opérations courantes :

- on connait la position de chaque sommet d'un triangle dans $(O, \vec{x}, \vec{y}, \vec{z})$.
- on connait la position et l'orientation de l'observateur dans $(O, \vec{x}, \vec{y}, \vec{z})$.
- quelle est la position des sommets du triangle dans le repère de l'observateur $(E, \vec{u}, \vec{v}, \vec{w})$?
- déterminer que les sommets du triangle sont visibles ?

Quelques précisions sur l'obervateur

- définir sa position et son orientation,
- définir sa "projection",
- et la résolution de l'image ?
- déterminer la direction d'un rayon passant par le centre du pixel (x, y) ?
- déterminer qu'un sommet est visible ?
- déterminer qu'une primitive (triangle) est visible ?

succession de transformations simples ...

Quelques précisions sur l'observateur

plusieurs repères :

- repère de l'objet (matrice *Modele* $\rightarrow$),
- repère de la scène (matrice *Vue* $\rightarrow$),
- repère de l'observateur (matrice *Projection* $\rightarrow$),
- repère de projection de l'observateur (matrice *Fenetre* $\rightarrow$),
- repère de l'image.

connaissant les matrices de passage, repère par repère :

- sur quel pixel se projette un sommet d'une primitive ?
- quelle direction dans le repère de la scène correspond à la direction du pixel (x, y) ?

Matrice de "projection"

pour simplifier / réaliser toutes les manipulations précédentes la matrice de "projection" doit être inversible ...

ce qui n'est pas le cas par définition.

et alors ?

trouver une transformation affine équivalente.

Matrice de "projection" orthographique

on se place dans un repère "canonique", un cube unitaire $[-1\ 1]^3$.

une "projection" orthographique conserve la taille des objets, un point (x, y, z) se projette sur le pixel (x, y).
et z reprèsente la distance du point.

transformation affine :
$T(\vec{x}) = \vec{x}$
$T(\vec{y}) = \vec{y}$
$T(\vec{z}) = \vec{z}$
$T(O) = O$

représentation : matrice identité !

Matrice de "projection" perspective

idée :

se ramener au cas de la "projection" orthographique ?

solution :

une pyramide de vision n'est finalement qu'un cube déformé !

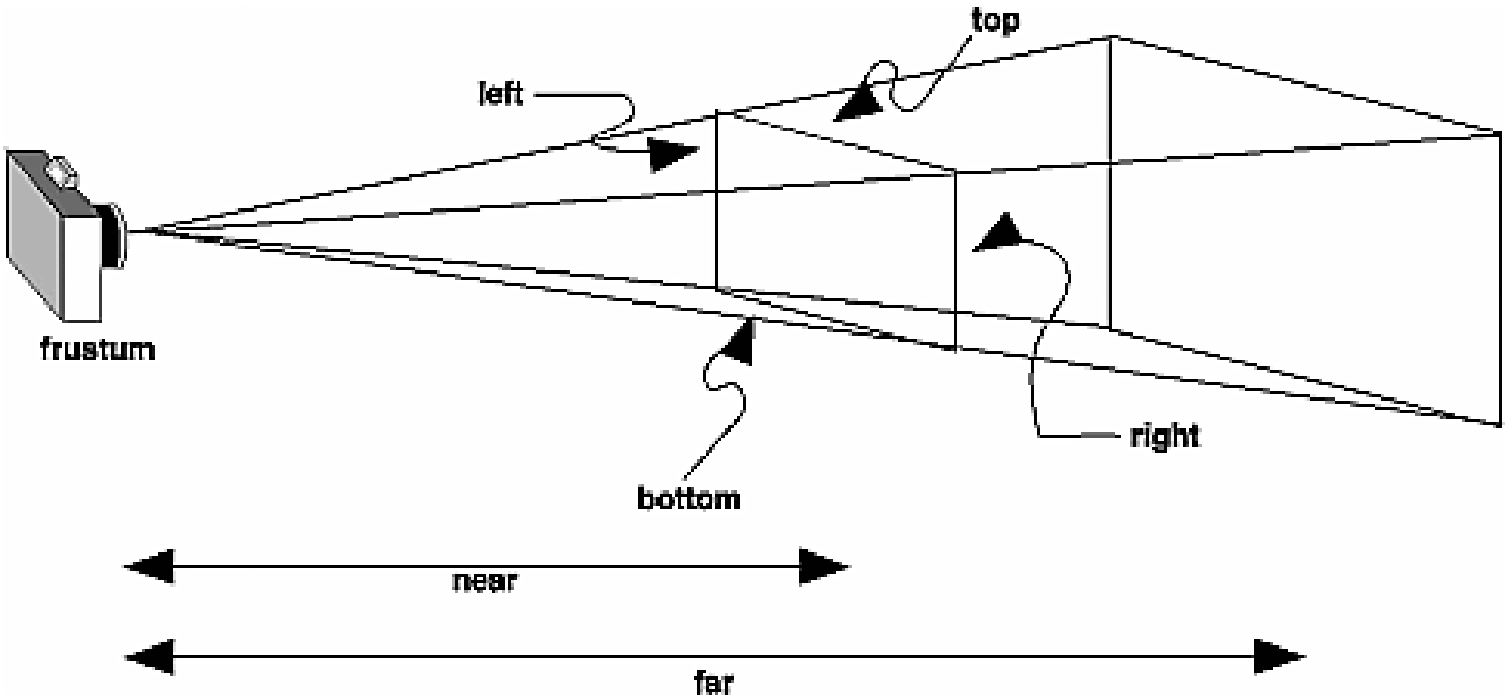
même transformation utilisée par OpenGL et DirectX.

Matrice de "projection" perspective

$$\begin{bmatrix} \frac{2n}{r-l} & 0 & -\frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & -\frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

transformation affine de la scène vers un cube unitaire $[-1\ 1]^3$.
volume de vision :
$z_{min} = near$ et $z_{max} = far$,
$x_{min} = left$, $x_{max} = right$,
$y_{min} = bottom$, $y_{max} = top$.

Matrice de "projection" perspective



Transformation "Fenetre"

dernière étape de la chaine de transformations : passer du cube unitaire $[-1\ 1]^3$ "projectif" aux coordonnées du pixel.

c'est à dire passer d'un intervalle $[-1\ 1]$ à $[0\ n]$
(pour $n =$ largeur, hauteur, profondeur $\equiv 1$).

composition d'une mise à l'échelle et d'une translation :
$[-1\ 1] \rightarrow [0\ 2] \rightarrow [0\ n]$

Transformation "Fenetre"

l'espace "*Fenetre*" n'est pas un plan, mais un cube $[0\ \mathit{largeur}] \times [0\ \mathit{hauteur}] \times [0\ 1]$.

créer un rayon dans ce repère et le transformer pour obtenir son origine et sa direction dans le repère de la scène (ou d'un objet).

$o_{Fenetre}(x, y) = (x, y, 0)^t$
$e_{Fenetre}(x, y) = (x, y, 1)^t$
$\vec{d}_{Fenetre}(x, y) = e_{Fenetre}(x, y) - o_{Fenetre}(x, y)$

Pipeline graphique et visibilité

rappels :

- les objets sont décrits par un ensemble de primitives,
- les sommets (ou les points de contrôle) des primitives sont connus dans un repère "local" à l'objet.
- "passer" les sommets dans le repère projectif,
- vérifier qu'ils sont visibles,
- déterminer les coordonnées du pixel correspondant (passage dans le repère fenêtre).

les pixels de l'image doivent avoir la couleur de l'objet le plus proche (visible à travers le pixel).

Pipeline graphique et visibilité

pour chaque objet :

- pour chaque primitive de l'objet :
- pour chaque sommet de la primitive :
- projetter le sommet :
  passage repère local $\rightarrow$ repère projection homogène,
- déterminer la visibilité du sommet.
- déterminer la visibilité de la primitive,
- déterminer l'ensemble de pixels couvrant la partie visible de la primitive,
- pour chaque pixel :
- déterminer sa couleur.

Visibilité

on dessine les objets dans un ordre "quelconque" :

pour obtenir une image correcte, il faut déterminer l'objet le plus proche visible à travers chaque pixel.

remarque :

- pour éviter de confondre un pixel en cours de calcul et un pixel déjà écrit dans l'image (et le z-buffer) :
- *fragment* : "pixel" en cours de calcul + élement de surface de la primitive visible + profondeur,
- *pixel* : fragment écrit dans l'image (et le z-buffer).

image de profondeur (ou z-buffer) conserve la profondeur de l'objet visible à travers chaque pixel.

Visibilité : algorithme

afficher plusieurs objets :

- pour chaque objet :
- ...
- pour chaque *fragment* :
- déterminer sa profondeur,
- n'écrire le *fragment* dans l'image et le z-buffer que si sa profondeur est inférieure à la profondeur déjà écrite dans le z-buffer.

Visibilité : détails

même algorithme que la recherche de la plus petite valeur d'un ensemble :

- min= max (valeur plus grande que les valeurs de l'ensemble)
- pour i $\in [0..n)$
- si ensemble[i] $<$ min
- min $\leftarrow$ ensemble[i]

quelle valeur utiliser pour la profondeur max ?

Visibilité d'un point

par définition :

- un point p est visible si sa projection est dans le cube $[-1..1]^3$ du repère projectif.

mais le repère projectif est homogène :

- $p_h = P(V(Mp))$

Visibilité d'un point

$$p_h = \begin{bmatrix} x_h \\ y_h \\ z_h \\ w_h \end{bmatrix}$$

est visible si :

$$\begin{aligned} -w_h < \quad x_h \quad < w_h \\ -w_h < \quad y_h \quad < w_h \\ -w_h < \quad z_h \quad < w_h \end{aligned}$$

alors p se projette dans le repère fenêtre : $p_f = F(p_h/w_h)$ et $p_f \in [0..largeur] \times [0..hauteur] \times [0..1]$

Visibilité d'une primitive

plusieurs sommets :

vérifier que tous les sommets sont visibles . . .

si ce n'est pas le cas :

- soit découper la primitive et retester chaque "morceau",
- soit trouver les *fragments* qui remplissent la partie visible de la primitive.

Résumé

schema.

et alors ?

plusieurs solutions pour déterminer les objets visibles.

solution 1 :

- projetter les sommets des primitives,
- redécouper les primitives partiellement visibles,
- *fragmenter* les primitives totalement visibles dans le repère Fenêtre.

solution 2 :

- projetter les sommets des primitives,
- *fragmenter* les primitives dans le repère projectif homogène.

et alors ?

plusieurs solutions pour déterminer les objets visibles.

solution 3 :

- déterminer l'équation de la droite passant par chaque pixel,
- trouver les objets intersectants la droite et dessiner le plus proche.

et alors ?

quelle solution est utilisée :

- par une carte graphique ?
- par un lancer de rayons ?
- par REYES ?

et avec des primitives non planes ?

et alors ?

solution utilisée :

- par une carte graphique : solution 2
- par un lancer de rayons : solution 3
- par REYES : généralisation de la solution 1 (primitives non planes).

et alors ?

REYES est une généralisation de la "fragmentation" homogène utilisée par les cartes graphiques.

différences :

- les primitives ne sont pas forcément planes,
- les primitives de modélisation ne sont pas les mêmes que celles utilisées pour la fragmentation,
- les primitives utilisées pour la fragmentation sont des quads de taille inférieure à un pixel (micro-polygones).
- on "redécoupe" les primitives tant qu'elles ne sont pas totalement visibles et que l'on ne peut pas générer "correctement" les micro-polygones.