Images : OpenGL

Corentin Ferry*

26 avril 1986

Содержание

1 Introduction à OpenGL

Cette section, un peu inutile pour l'évaluation, est à écrire en dernier. Éventuellement, on peut insérer une petite explication sur le principe des fonctions C qu'OpenGL implémente (`glSphered` par exemple...).

2 Modélisation

2.1 Modélisations surfacique, volumique

Modélisation volumique La modélisation volumique, passons très rapidement dessus. Elle n'est pas utilisée dans le cadre du rendu 3D, mais plutôt en CAD.

Un volume se décrit par exemple géométriquement par induction, avec :

- L'ensemble de base constitué de volumes simples, du style sphères, parallélépipèdes rectangles, etc, que l'on sait construire dans le monde du discret;
- Des fonctions d'assemblage comme l'union, l'intersection, la différence symétrique...

On appelle ceci de la **constructive solid geometry**.

Si on travaille en discret, on peut aussi représenter un volume par des **voxels**, ce qui signifie "volume pixels à condition d'avoir subdivisé l'espace en petits cubes. On choisit alors de quel matériau un cube est rempli.

*cferr@openmailbox.org

```
fc1 = (, $P_1$, $P_2$);
fc2 = ($P_1$, $P_2$, $P_3$);
fc3 = ($P_2$, $P_3$, $P_4$);

surf = (fc1, fc2, fc3);
```

```
fc1 = (, $P_1$, $P_2$);
fc2 = ($P_1$, $P_2$, $P_3$);
fc3 = ($P_2$, $P_3$, $P_4$);

surf = (fc1, fc2, fc3);
```

Рис. 1: Deux codes, qui montrent ce qu'il faut faire et ce qu'il ne faut pas faire

Modélisation surfacique Lorsque l'on travaille sur des cartes graphiques : on n'a pas besoin de plonger à l'intérieur d'un objet que l'on voit de l'extérieur (la plupart du temps), donc une modélisation surfacique suffit. Cela évite un nombre assez conséquent de calculs inutiles.

On représente les surfaces de manière discrète, par des polygones, lesquels peuvent être représentés par des **triangles** (cf. Delaunay), lesquels ne sont jamais que des listes de points. En d'autres termes, il suffit d'avoir une liste de points suffisamment bien ordonnée pour représenter une surface.

L'ordre de la liste de points induit aussi **l'orientation de la surface** (celle dont on déduit le vecteur normal). Cette définition est implicite.

Une carte graphique ne sait en fait manipuler que des triangles. Il va de soi que la qualité du rendu dépend du nombre de triangles en jeu.

Partage des points dans la construction des facettes Intuitivement, on va s'éviter de dupliquer les points utilisés pour la construction d'une surface. Ainsi, on préfèrera le code de droite au code de gauche.

2.2 Interpolation de surfaces

Courbes de Bézier Déjà, si la question est posée : Bézier n'était pas un mathématicien ! C'était un ingénieur chez Citroën. Pour que Citroën puisse produire de belles caisses (comment voulez-vous vendre ?), il inventa des courbes à base de polynômes de Bernstein (Сергей Натанович Бернштейн), russe[1].

Un petit rappel de la formule du k-ième polynôme de degré n de Bernstein :

$$B_k^n(x) = \binom{n}{k} x^k \, (1-x)^{n-k} \text{ pour } x \in [0,1]$$

Une surface est alors définie par la double somme de Bézier :

$$S(x,y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} P_{i,j} B_i^{n_x}(x) B_j^{n_y}(x) \text{ pour } x,y \in [0,1]$$

où les P_i, j sont les points de passage dits **points de contrôle**. Il faut faire l'homothétie pour transformer ça en une surface définie ailleurs que sur $[0,1]^2$.

Lors du rendu, la surface de Bézier est discrétisée à un certain pas, dont dépend le nombre de triangles fabriqué, et donc le temps de rendu.

3 Rendu

4 Animation

[1] Je précise qu'il a étudié à Supélec et à la Sorbonne !