

TP OpenGL 5 - Extrusion

Licence Informatique 3ème année

Année 2016-2017

1 Introduction

L'objectif de ce Tp est d'implémenter une fonction permettant de construire une surface 3D par extrusion d'un contour 2D. Pour ce Tp on donne la structure de données suivante, qui permet de représenter en mémoire un point de l'espace 3D :

```
typedef struct {  
    float x, y, z; /* coordonnées 3D du point */  
} point3D;
```

Les différentes versions de la fonction d'extrusion à écrire devront être réalisées dans le cadre d'une application permettant la visualisation de l'objet construit, tout en offrant les fonctionnalités suivantes :

- zoom avant et arrière
- rotation autour des axes Ox et Oy

Ces fonctionnalités auront pour objectif de permettre la visualisation du l'objet sous toutes ses faces. Il vous est conseillé de partir de l'une des applications développées dans les TP précédents.

Exercice 1

Ecrire le code, en langage C et en utilisant certaines fonctions OpenGL, de la fonction suivante :

```
void extrude(point3D t, point3D contour[], int nbsom)
```

avec

- **t** un point en 3D, qui représente une translation ;
- **contour** un tableau de points en 3D, qui représentent le contour d'un polygone ;
- **nbsom** le nombre de points présents dans **contour**.

Cette fonction doit générer l'affichage des primitives géométriques qui permettent de représenter un objet issu de l'**extrusion** du contour en suivant le vecteur translation. La figure 1 illustre le fonctionnement de cette fonction.

On précise les points suivants :

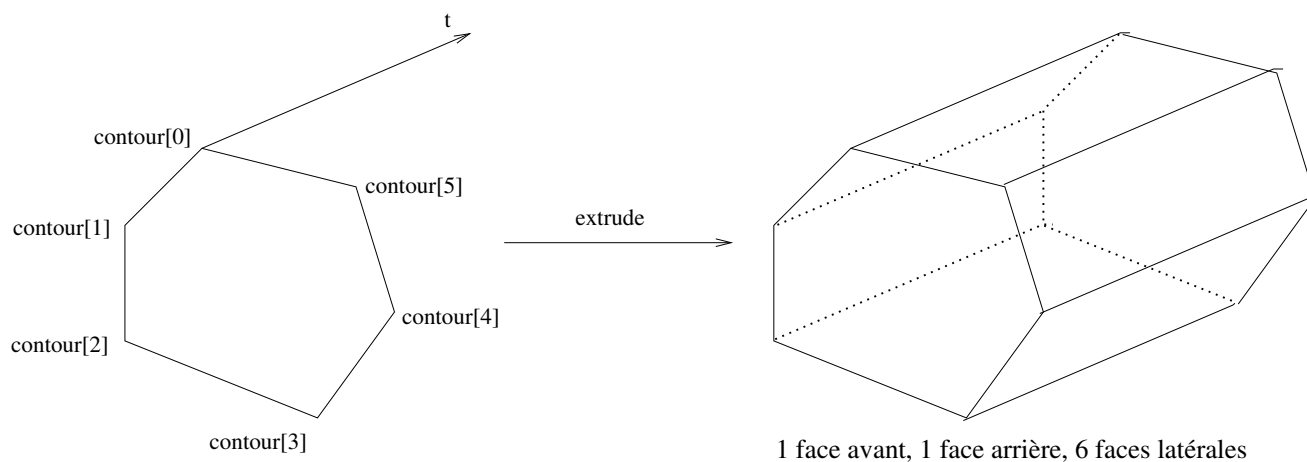


FIGURE 1 – Obtention d'un volume par extrusion d'un polygone

- le **contour** n'est pas « fermé », au sens où le point d'indice 0 et le point d'indice $nbsom - 1$ ne sont pas égaux ;
- les faces doivent être générées dans le sens **direct**, c'est à dire en suivant les règles dites du « tire-bouchon » ou des « trois doigts » ;
- les faces avant et arrière du volume doivent être affichées ;
- les paramètres **t** et **contour** ne doivent pas avoir été modifiés à la sortie de la fonction.

Question 2

Modifier la fonction **extrude** de manière à ce que son prototype corresponde à celui qui est présenté ci-après :

```
void extrude(point3D t, point3D contour[], int nbSom,
            int faceAvant, int faceArriere)
```

avec

- **t** un point en 3D, qui représente une translation ;
- **contour** un tableau de point en 3D, qui représentent le contour d'un polygone ;
- **nbSom** le nombre de points présents dans **contour** ;
- **faceAvant** qui vaut 1 si la face avant du contour extrudé doit être affichée (ce qui était le cas dans la version précédente) et 0 sinon ;
- **faceArriere** qui vaut 1 si la face arrière du contour extrudé doit être affichée (ce qui était le cas dans la version précédente) et 0 sinon.

Exercice 3

Modifier la fonction **extrude** de manière à ce que son prototype corresponde à celui qui est présenté ci-après :

```
void extrude(point3D t[], int nbt, point3D contour[], int nbsom,
            int faceAvant, int faceArriere)
```

avec

- **t** un tableau dont chaque case représente une translation ;
- **nbt** le nombre de cases dans **t** ;
- **contour** un tableau de points en 3D, qui représentent le contour d'un polygone ;
- **nbsom** le nombre de points présents dans **contour** ;
- **faceAvant** qui vaut 1 si la face avant du contour extrudé doit être affichée et 0 sinon ;
- **faceArriere** qui vaut 1 si la face arrière du contour extrudé doit être affichée et 0 sinon.

Cette fonction le volume obtenu par **extrusions successives** du contour polygonal. Son fonctionnement est illustré sur la figure suivante.

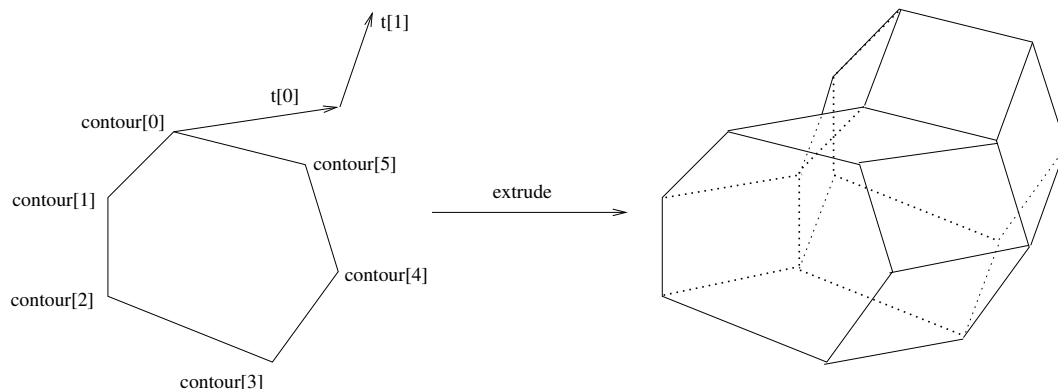


FIGURE 2 – Volume obtenu par 2 extrusions successives d'un polygone