

TP 1 : Maillages

Exercice 1 : Structure de données *Half-Edge*

Définissez en C++ une structure de données *Half-Edge* pour la gestion de maillages polygonaux. Pour cela, définissez les classes `Vertex`, `HalfEdge`, `Face` et `Mesh` (attributs, constructeurs/destructeurs, accesseurs, ...).

Rappel des attributs :

- `Vertex` : identifiant (entier) + coordonnées + pointeur vers **une** demi-arête incidente (dont il est source);
- `HalfEdge` : pointeur vers le sommet source + pointeur vers la face à laquelle elle appartient + pointeurs vers la demi-arête précédente et la demi-arête suivante dans la face + pointeur vers la demi-arête tête-bêche;
- `Face` : identifiant (entier) + pointeur vers **une** demi-arête incidente ;
- `Mesh` : ensemble des sommets + ensemble des faces.

Exercice 2 : Chargement et sauvegarde de maillages

Ajoutez à la classe `Mesh` une méthode `import()` qui charge un maillage polygonal fourni dans un fichier au format `OFF`. Le format `OFF` commence par le mot-clé `OFF`, puis une ligne `x y 0` avec `x` et `y` respectivement le nombre de sommets et de faces du maillages, puis `x` lignes avec les coordonnées de chaque sommet, puis `y` lignes avec le nombre et les indices des sommets de chaque face (les indices vont de 0 à `x-1`).

Indication : pour faciliter le calcul de la demi-arête tête-bêche à une demi-arête, vous pouvez utiliser une table de hachage (`map<int, vector<HalfEdge*> >`) sur les identifiants des sommets, qui stocke pour chaque sommet les demi-arêtes dont il est la source.

Ajoutez à la classe `Mesh` une méthode `export()` qui sauvegarde un maillage polygonal donné sous forme de structure *Half-Edge* dans un fichier au format `OFF`.

Testez vos méthodes `import()` et `export()` grâce aux fichiers `OFF` fournis.

Le TP est à terminer pendant votre temps libre d'ici le prochain TP, car vous aurez besoin de ces structures et méthodes pour le TP 2.