

TP 2 – Surfaces de subdivision et structure Half-Edge

Partie I - Implémentation d'une structure Half-Edge en Python et chargement de maillages 3D

Objectif : Le but de cette partie du TP est de :

- Comprendre et implémenter pas à pas une structure Half-Edge pour représenter un maillage polygonal en 3D.
- Lire un fichier de maillage (format .off ou .obj simple) pour construire et tester votre structure Half-Edge.
- Vérifier la cohérence du maillage et afficher les informations principales.

Structure Half-Edge

La structure Half-Edge permet de représenter efficacement un maillage en gardant une information complète sur :

- Les sommets,
- Les facettes,
- Les arêtes (chaque arête est représentée par deux half-edges (demi-arêtes) opposés,
- Et la connectivité locale (next, prev, twin(opposé)).

Cette structure est particulièrement adaptée aux applications en traitement de maillages 3D : simplification, subdivision, recalcul de normales, etc.

Pour mieux comprendre cette structure, je vous invite à consulter : <https://jerryin.info/geometry-processing-algorithms/half-edge/>

Question 1 - Expliquer la structure demi-arête (half-edge) en utilisant un exemple (sans coder)

Question 2 - Implémenter les classes suivantes en Python :

- **Vertex :**
 - Coordonnées 3D (x, y, z)
 - Référence à une **half-edge sortante**.
- **HalfEdge :**
 - Référence vers le **vertex cible** (destination).
 - Références vers :
 - Le **twin** (half-edge opposé),
 - Le **next** (suivant dans la face),
 - Le **prev** (précédent dans la face),
 - La **facette** associée.
- **Face :**
 - Référence vers **une half-edge appartenant à la face**.
- **Mesh :**
 - Contient les **listes de Vertex, HalfEdge et Face**.
 - Méthodes principales :
 - Lecture du fichier .off ou .obj
 - Construction du graphe Half-Edge

- Affichage des propriétés principales (nombre de sommets, faces, arêtes).

Question 3

1. Lire un fichier .off (ou .obj simple) en utilisant la structure Half-Edge de la question 2.
2. Afficher :
 - Le nombre total de **sommets**, **facettes** et **half-edges**.
 - Pour quelques sommets :
 - La liste des **voisins immédiats** (sommets directement connectés).
 - Pour quelques facettes :
 - La liste des **sommets formant la facette**.

Question 4

- Détecter automatiquement les **arêtes de bord** (qui n'ont pas de twin).
- Calculer la **normale** d'une face.
- Colorier les **faces frontières** différemment lors de la visualisation.

Pour tester votre code vous pouvez utiliser un exemple de fichier .off simple (Deux triangles formant un carré)

```
OFF
4 2 0
0 0 0
1 0 0
1 1 0
0 1 0
3 0 1 2
3 0 2 3
```

Question 5 . Le github suivant traite la problématique ci-dessus.

https://github.com/carlosrojas/halfedge_mesh

Établir la matrice SWOT de ce code (Les forces, Les faiblesses, Opportunités et Risques).

Partie II – Surfaces de subdivision

Exercice 1. Subdivision de Loop

1. Rappeler le principe de la subdivision de Loop
2. On considère un maillage manifold fermé (sans bord).
 - a. Écrire une fonction Python qui implémente le schéma de subdivision de Loop pour ce type de maillage.
 - b. Afficher le résultat de subdivision pour différentes itérations.
3. On considère maintenant un maillage manifold avec bord.
 - a. Que faut-il modifier pour appliquer le schéma de subdivision de Loop ?
 - b. Mettre à jour la fonction Python de la question 2. pour tenir compte des maillages avec bord.

- c. Afficher le résultat de subdivision sur **un maillage avec bord** pour différentes itérations.

Exercice 2. Subdivision Butterfly Modifiée

1. Rappeler le principe de la subdivision **Butterfly modifiée**
2. On considère un maillage manifold fermé (sans bord).
 - a. Rappeler la notion de valence d'un sommet. Qu'est-ce qu'un maillage régulier ?
 - b. Écrire une fonction Python qui implémente le schéma de **Butterfly modifiée** pour ce type de maillage.
 - c. Afficher le résultat de subdivision pour différentes itérations.
3. On considère maintenant un maillage manifold avec bord.
 - a. Que faut-il modifier pour appliquer le schéma de **Butterfly modifiée** ?
 - b. Mettre à jour la fonction Python de la question 2. pour tenir compte des maillages avec bord.
 - c. Afficher le résultat de subdivision sur **un maillage avec bord** pour différentes itérations.

Livrables et deadlines

Équipe : Le travail est par groupe de **2 étudiants max.**

Livrables :

1. **Si** vous utilisez Jupiter notebook **alors Un notebook détaillé** (avec une exportation en html, permettant de voir tout le travail fourni sans être contraint à exécuter le notebook).
2. **Sinon** : Code en Python du TP + Un rapport succinct (max 3 pages) qui détaille le déroulement et présente les résultats

Deadline : 19 mai 2024 à 23h00 (**aucun retard n'est autorisé**).

Références

[1] M.S. Mathematics, University of Utah. 1987. Thesis: ["Smooth Subdivision Surfaces Based on Triangles"](#).

[2] D. Zorin, et. al. "Interpolation Subdivision for Meshes with Arbitrary Topology", 1996. (<https://mrl.cs.nyu.edu/~dzorin/papers/zorin1996ism.pdf>)

[3]

Remarque.

Vous pouvez utiliser ChatGPT. Dans ce cas il faudra le signaler et préciser les questions posées et les différentes interactions effectuées.