

Résumé itération 2

Cette itération a été la continuité de l'implémentation de la structure de données, notamment son perfectionnement au niveau des performances, et l'implémentation des premiers liens entre le visuel 3D et cette structure de données.

Fonctionnalités implémentées :

1. Amélioration de la rapidité du remplissage des données avec un nouvel algorithme (et résolution de bugs). (*Tanguy*)
2. Fonctionnalité d'ajustement de la caméra pour placer l'objet 3D au centre en double-cliquant dessus. (*Xin*)
3. Fonctionnalité d'export de la structure de données en fichier STL ou OBJ. (*Tanguy*)
4. Ajout du repère dans le plan. (*Xin*)
5. Implémentation de la fonctionnalité permettant de changer les coordonnées d'un point depuis le menu de modification ou depuis l'objet 3D (lorsqu'une face est sélectionnée et que les sommets sont surlignés). (*Lucie*)
6. Implémentation de la fonctionnalité permettant de passer d'une visualisation de l'objet 3D en "fil de fer" ou en "matériel". (*Solène*)
7. Fonctionnalité de configurer l'anti-aliasing. (*Xin*)

Changements réalisés par rapport à l'étude préalable :

- L'algorithme de détection des trous a été implémenté durant l'itération 1, il n'a pas encore été testé durant cette itération.
- La fonctionnalité 2 a été rajoutée.
- La fonctionnalité 3 était initialement prévue pour l'itération 5.
- La fonctionnalité 4 n'était pas prévue.
- La fonctionnalité 7 n'était pas prévue.

Changement réalisés par rapport à la préparation de l'itération 2 à la fin de l'itération 1 :

- On a jugé que la fonctionnalité permettant de personnaliser la couleur de l'objet 3D et la couleur de la face sélectionnée n'était pas importante à ce stade du projet et a été reportée à la fin du projet.
- L'algorithme de remplissage des données a été remanié pour le rendre plus rapide. Une structure de données de type SkipList a été implémentée pour optimiser la recherche de sommets déjà existant à la création des sommets du modèle 3D.

Nouvel algorithme de remplissage des données :

```
fonction convertSTLToData (positions : Tableau[0...MAX_TAB]) : Mesh
début
sommets ← new VertexSkipList() //Nous permettra de retrouver plus rapidement un sommet

faces ← [ ] // tableau de faces du modèle 3D

pour i de 0 à MAX_TAB -1 avec un pas de 9 faire

    //On crée chaque sommet à partir des 3 triplets de coordonnées
    sommetsCourants ← [ ]
    sommetsCourants[1]← créerSommet( Point( positions[i], positions[i+1],
positions[i+2], sommets))
    sommetsCourants[2]← créerSommet( Point( positions[i+3], positions[i+4],
positions[i+5], sommets))
    sommetsCourants[3]← créerSommet( Point( positions[i+6], positions[i+7],
positions[i+8], sommets))

    halfedges ← [ ] //Tableau d'halfedges du modèle 3D

    pour chaque sommet de sommetsCourant faire
        he ← new Halfedge(sommet)
        halfedges.push(h)
    fpour

    pour j de 0 à sommetCourant.length faire
        sommet ← sommet[i]
        halfedge ← halfedges[i]
        sommet.addHalfedge(halfedge)
    fpour

    face ← new Face(halfedges[0])
fpour
halfEdgeProbleme ← sommet.getHalfEdgeProblem()
mesh ← new Mesh(faces, halfEdgeProbleme)
```

retourne mesh

fin

Lexique :

- **sommets** : VertexSkipList, structure de données dans laquelle sont stockées les nouveaux sommets. Elle permet une recherche dichotomique des sommets
- **faces** : Tableau de faces, tableau dans lequel sont stockées les faces du modèles 3D
- **sommetsCourant** : Tableau de sommets, tableau dans lequel sont stockées les 3 sommets de la face en cours de création
- **halfedges** : Tableau de HalfEdge, tableau dans lequel sont stockées les HalfEdges de la face en cours de création
- **face** : Face, la face en cours de création

- **Face <edge : HalfEdge>** : Représente une Face du modèle 3D
- **Vertex <point : Point, halfEdgeTab : tableau de HalfEdge>** : Représente le sommet se trouvant au coordonnées *point*. Il possède également la liste des HalfEdge partant de celui-ci
- **HalfEdge<vertex : Vertex, face : Face, next : HalfEdge, prev : HalfEdge, opposite : HalfEdge>** : Représente une arête d'une face. Elle connaît la face à laquelle elle appartient, son sommet de départ, ainsi que l'halfedge qui la précèdent, qui la suit et son opposée.

fonction créerSommet(point : Point, sommets : VertexSkipList()) : Vertex

début

existingVertex ← sommets.search(point)

si existingVertex = null **alors**

 existingVertex = new Vertex(point)

 sommets.insertVertex(point, existingVertex)

fsi

retourne existingVertex

fin

Lexique :

- **existingVertex** : Vertex, le sommet à vérifier s'il existe déjà, sinon on le crée
- **point** : Point, le point de départ du sommet considéré
- **sommets** : VertexSkipList, structure de données dans laquelle sont stockées les nouveaux sommets. Elle permet une recherche dichotomique des sommets

Dans la classe Vertex :

fonction addHalfEdge(he : HalfEdge)

début

//tailVertex permet de récupérer le sommet d'arrivée de l'halfedge et halfedgeTab est un tableau contenant toutes les halfedge partant de ce sommet

arrivée = he.tailVertex()

//On recherche dans le sommet d'arrivée de he, une halfedge qui a comme point d'arrivée le sommet courant

opposée ← arrivée.halfedgeTab.find(halfedge => halfedge.tailVertex === this)

si l'opposée est trouvée **alors**

 he.setOpposite(opposée)

 opposée.setOpposite(he)

 Ensuite on retire *opposite* du sommet d'arrivée

sinon

 On ajoute *he* dans *this.halfedgeTab*

fsi

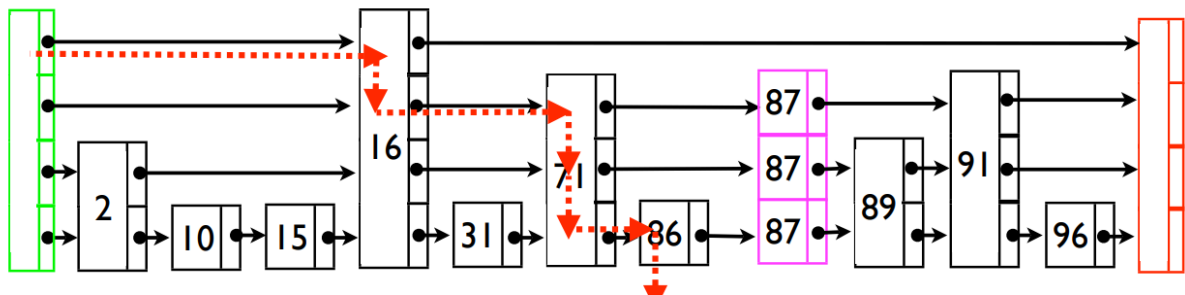
fin

Lexique :

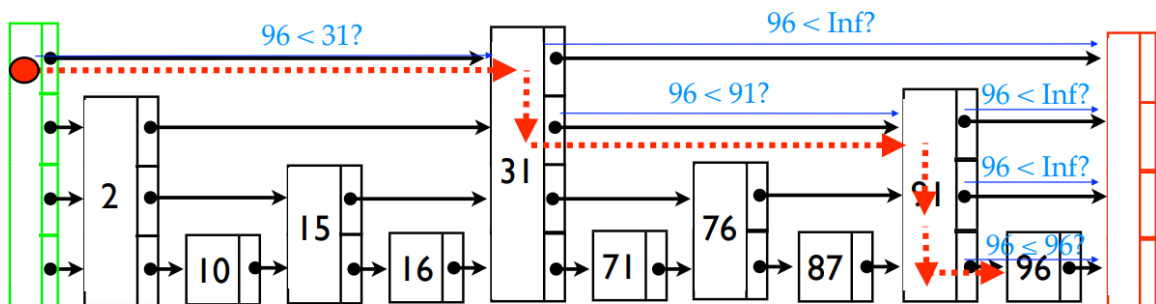
- he : HalfEdge, l'halfEdge considérée
- arrivée : Vertex, le sommet d'arrivée de l'halfEdge considérée
- opposée : HalfEdge | null, l'opposée de l'halfEdge considérée, si elle existe

Schéma SkipList :

Insert 87



Find 96



Préparations par rapport à l'itération 3 :

Fonctionnalités prévues :

- Fonctionnalité permettant de sélectionner plusieurs faces du modèle 3D (indépendamment de la structure de données).
- Tester les algorithmes de détection des trous sur des petits objets.
- Début des algorithmes de réparation du maillage.
- Fonctionnalité de communication d'erreurs.
- Fonctionnalité d'affichage des arêtes à erreur.
- Fonctionnalité de génération d'un point et des trois triangles associés.