

**fonction convertirSTLToData(geometry : BufferGeometry) : Mesh**

**début**

```
sommets ← []
faces ← []
positions ← geometry.position
pour i de 0 à positions.length avec pas de 9
    sommet 1 ← créer sommet1(point(triplet1), sommets) //un triplet et une
    succession de 3 éléments du tableau (x,y,z)
    sommet2 ← créer sommet2(point(triplet2), sommets)
    sommet3 ← créer sommet3(point(triplet3), sommets)
    h1{départ : sommet1, arrivée : sommet2} //une halfedge avec un sommet de départ et un
    d'arrivée
    h2{départ : sommet2, arrivée : sommet3}
    h3{départ : sommet3, arrivée : sommet1}
    h1.next ← h2
    h1.prev ← h3
    h2.next ← h3
    h2.prev ← h1
    h3.next ← h1
    h3.prev ← h2
    sommet1.he.add(h1,h3)
    sommet2.he.add(h1,h2)
    sommet3.he.add(h2,h3) // he à supprimer à la fin ?
    face ← créer Face(h1)
    faces.add(face)
fpour
pour chaque face de faces
    pour chaque halfedge de face
        trouverOpposée(halfedge)
    fpour
fpour
mesh ← {faces, sommets} //représentation du modèle 3D
retourne mesh
```

**fin**

### Lexique

*geometry* : BufferGeometry → Geometry de type BufferGeometry (élément de la librairie three.js)

*sommets* : tab[Sommet] → tableau contenant les sommets du fichier STL

*faces* : tab[Face] → tableau contenant les faces du fichier STL

*positions* : tab[Float] → tableau contenant les coordonnées x, y et z du BufferGeometry retourné par le STLLoader de three.js

*i* : Int → indice d'itération de la boucle

*sommet1* : Sommet → sommet créé pour la représentation du fichier STL en maillage (triangles)

*sommet2* : Sommet → sommet créé pour la représentation du fichier STL en maillage (triangles)

*sommet3* : Sommet → sommet créé pour la représentation du fichier STL en maillage (triangles)

*h1* : HalfEdge → HalfEdge créé pour la représentation du fichier STL en maillage (triangles), partant du *sommet1* au *sommet2*

*h2* : HalfEdge → HalfEdge créé pour la représentation du fichier STL en maillage (triangles), partant du *sommet2* au *sommet3*

*h3* : HalfEdge → HalfEdge créé pour la représentation du fichier STL en maillage (triangles), partant du *sommet3* au *sommet1*

*face* : Face → Face créé pour la représentation du fichier STL en maillage (triangles), ayant pour base *h1*

*mesh* : Mesh → représentation du modèle 3D

---

### fonction trouverOpposée(halfedge : HalfEdge) : HalfEdge

début

*sommetDepart* ← halfedge.depart //sommet d'arrivée de halfedge

*sommetArrive* ← halfedge.arrivee

    // les points de départ et d'arrivée de l'halfEdge opposée doivent être inversé avec l'halfedge courante

*opp* ← départ.halfedge.filter( h → h.arrivée === *sommetDepart* && h.départ === *sommetArrive*)

    si *opp* différent de 'undefined'

        alors si *opp.opposite* non nul

            alors Erreur('Halfedge opposé déjà défini')

        sinon

            halfedge.setOpposite(*opp*)

*opp.setOpposite*(halfedge)

    fsi

fin

fin

### Lexique

*halfedge* : HalfEdge → l'HalfEdge pour laquelle on cherche son HalfEdge opposée

*sommetDépart* : Sommet → sommet de départ de l'HalfEdge opposée, correspondant au sommet d'arrivée de *halfedge*

*sommetArrive* : Sommet → sommet d'arrivée de l'HalfEdge opposée, correspondant au sommet de départ de *halfedge*

*opp* : HalfEdge → l'HalfEdge opposée à *halfedge*

---

**fonction créerSommet(point : Point, tabSommets : Tableau de sommets)**

**debut**

```
// chaque sommet doit avoir une instance unique dans le tableau
sommet ← new Sommet(point)
existSommet = tabSommets.find(vertex => vertex.point.equals(point))
si existSommet egal a 'undefined'
    alors ajouter(sommet, sommets)
    sinon sommet ← getSommetInList(sommet, sommets)
```

**fsi**

**retourne** sommet

**fin**

### **Lexique**

*point* : Point → point à partir duquel on souhaite créer un sommet

*tabSommets* : tab[Sommet] → tableau contenant tous les sommets du fichier STL

*sommet* : Sommet → nouveau sommet créé par cette fonction à partir du point donné

*existSommet* : Boolean → si vrai, le sommet voulant être créé existe déjà dans le tableau de sommets (donc dans la structure)