**Algorithme** remplissageDesDonnees

**Fonction** convertSTLToData (positions : **Tableau[0…n]**) : **Mesh**

**début**

1.   sommetsSkipList ← new VertexSkipList()

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

3.   **pour** i **de** 0 **à** n -1 **avec un pas** de 9 **faire**

4.      sommetsCourants ← [ ]

5.      sommetsCourants[1] ← créerSommet(new Point( positions[i], positions[i+1],positions[i+2, sommetsSkipList])

6.      sommetsCourants[2]← créerSommet(new Point( positions[i+3], positions[i+4], positions[i+5, sommetsSkipList])

7.      sommetsCourants[3]← créerSommet(new Point( positions[i+6], positions[i+7], positions[i+8], sommetsSkipList)

8.       halfedges ← []

10.**pour chaque** sommet **de** sommetsCourant **faire**

11.           he ← new Halfedge(sommet)

12.           halfedges.push(h)

13.      **fpour**

14.      **pour** i **de** 0 **à** halfedges.length **faire**

15.           he ← halfedges[i]

16.           setPrevAndNext( he, halfedges[(i+2)%3], halfedges[(i+2)%3] )

17.      **fpour**

18.      **pour** j **de** 0 **à** sommetCourant.length **faire**

19.   sommet ← sommetCourant[i]

20.   halfedge ← halfedges[i]

21.   sommet.addHalfedge(halfedge)

22.     **fpour**

23.     face ← new Face(halfedges[0])

24.       **pour chaque** halfedge **de** halfedges **faire**

               halfedge.setFace(face)

25.       **fpour**

26.       faces.push(face)

27.    **fpour**

28.    halfEdgeProbleme ← sommet.getHalfEdgeProblem( )

29.    mesh ← new Mesh(faces, halfEdgeProbleme)

30.    **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 précède, qui la suit et son opposée.

**Fonction** créerSommet(point : Point, sommets : VertexSkipList) : Vertex

**début**

1. existingVertex ← sommets.searchVertex(point)

2.si existingVertex = null alors

3.    existingVertex = new Vertex(point)

4.    sommets.insertVertex(point, existingVertex)

**5.** fsi

**6.** retourne exisitingVertex

**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