Paul Mallard & Pauline Dos Santos E4FI group 2

Question 4 : Given a **myMesh** M, and a **myHalfedge** e, give the pseudo-code to collapse e into a vertex. So the two vertices of e are merged into a single vertex. There might be duplicate edges going to this new vertex, and you should remove any such duplicates.

```
myVertex* collapse(myMesh* M, myHalfedge* e){
   myHalfedge* e_twin = e->twin;
   myVertex* v1 = e->source;
   myVertex* v2 = e_twin->source;

   //new vertex in the middle position
   Vector3 mid_position = (v1->position + v2->position) / 2.0f;
   myVertex* v_new = M->addVertex(mid_position);

   //redirect halfedges from v1 or v2 to v_new
   for (myHalfedge* h : M->halfedges){
      if (h->source == v1 || h->source == v2){
         h->source = v_new;
      }
   }

   //faces
   if (e->adjacent_face != nullptr) {
      if (e->prev) e->prev->next = e_twin->next;
      if (e_twin->next) e_twin->next->prev = e->prev;
   }

   if (e_twin->adjacent_face != nullptr) {
      if (e_twin->prev) e_twin->prev->next = e->next;
      if (e->next) e->next->prev = e_twin->prev;
   }


   //delete
   M->removeHalfEdge(e);
   M->removeHalfEdge(e_twin);
   M->removeVertex(v1);
   M->removeVertex(v2);
   std::set<std::pair<int, int>> seen;
   std::vector<myHalfedge*> toDelete;

   for (myHalfedge* h : M->halfedges){
      myVertex* from_v = h->source;
      myVertex* to_v = h->next->source;

      if (from_v == v_new || to_v == v_new){
         int id1 = std::min(from_v->id, to_v->id);
```

```
            int id2 = std::max(from_v->id, to_v->id);
            std::pair<int, int> key = {id1, id2};

            if (seen.find(key) != seen.end()){
                if (h->twin != nullptr){
                    toDelete.push_back(h->twin);
                }
                toDelete.push_back(h);
            } else {
                seen.insert(key);
            }
        }
    }

    //delete halfedges
    for (myHalfedge* h : toDelete) {
        if (h->twin != nullptr) {
            h->twin->twin = nullptr; // couper les liens doubles
        }
        M->removeHalfEdge(h);
    }

    return v_new;
}
```