

out: April 7, Friday

due: **Apr. 19, Wednesday 10:00PM.** No extension!

This assignment is about the data structure of a triangle mesh. In particular, you are asked to implement the Half-Edge data structure that we discussed in class. You will also implement a few functions to manipulate the Half-Edge mesh.

1 About Starter Code

Our starter code is built on the code of programming assignment 1. The files that you need to finish with your code are under the `c2g2/geometry` folder.

2 Programming Requirements

You need to implement the following functions:

1. **Half-Edge data structure:** Please implement your Half-Edge data structure in the files `HalfEdge.java`, `Face.java`, and `Vertex.java`. Your implementation should follow what we discussed in class.
2. **Constructing a Half-Edge mesh:** Implement the constructor `HalfEdgeMesh(Mesh mesh)` that takes a indexed triangle mesh, one that you have used in your programming assignment 1, and converts it into a `HalfEdgeMesh`, one that use the Half-Edge data structure to store all the mesh data. For example, in the `HalfEdgeMesh` class, you need to maintain the following:
3. **Converting a Half-Edge mesh into an Indexed Triangle mesh:** Implement the method `HalfEdgeMesh.toMesh()` that converts the current `HalfEdge` mesh into an `Indexed Triangle` mesh that can be displayed using the code that you have implemented in PA-1.
4. **Removal of an vertex:** Implement the method `HalfEdgeMesh.removeVertex(Vertex vtx)`. The method removes the given vertex from a `HalfEdgeMesh`, and **fills the hole** left by that vertex. The expected result is illustrated in Figure 1.
5. **Removal of an edge:** Implement the method `HalfEdgeMesh.collapseEdge(HalfEdge edge)`. This method collapses a given edge into a vertex. Suppose an edge connecting vertex v_1 and v_2 . When this edge is collapsed, the edge is removed from the mesh and a new vertex $(v_1 + v_2)/2$ is added. The expected result is illustrated in Figure 2.

2.1 Visualizing Mesh Update

We provide the code to allow you to visualize the update of the mesh after an edge and a vertex are removed. After you implement the required methods (as specified above), you can press “n” to remove a vertex that we have hard coded and view the updated mesh. Similarly, you can press “m” to collapse an edge that we have hard coded and visualize the resulting mesh.

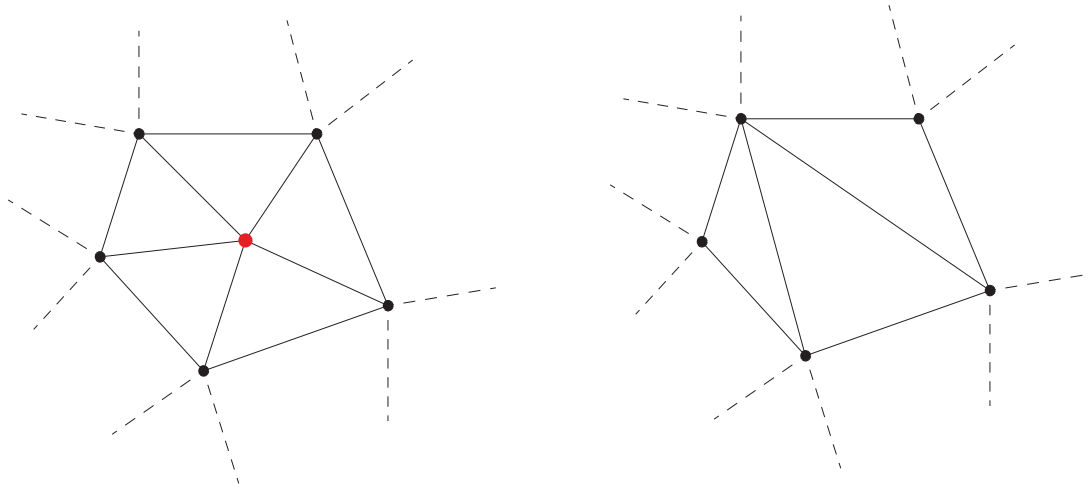


Figure 1: **Removal of a vertex.** Here the figures illustrate the expected change of a triangle mesh after removal of a vertex. The **left** figure shows the triangle mesh before vertex removal. The red vertex is about to be removed. The **right** figure shows the resulting mesh after the red vertex is removed. Note that the hole formed by the removal of the vertex is filled, and no open boundary is produced. Please implement `HalfEdgeMesh.removeVertex(...)` correctly to update the HalfEdge mesh as shown in the right figure.

For debugging purposes, you might want to explore writing your own visualization code to better visualize the topology of the mesh.

3 Submission and FAQ

Submission Checklist: Submit your assignment as a zip file via courseworks. Your submission must consist of the following parts:

1. **Documented code:** Your code should be reasonably documented and be readable/understandable by the TAs. If the TAs are not able to run and use your program, they will be unable to grade it. To ensure the TAs can grade your assignment, please make sure your code can be compiled successfully. It is your responsibility to make sure the code can be correctly compiled. Otherwise, you'll lose some points.
2. **Brief report:** Include a description of what you've attempted, special features you have implemented, and any instructions on how to run/use your program. In compliance with Columbia's Code of Academic Integrity, please include references to any external sources or discussions that were used to achieve your results.
3. **Screen capture of the mesh:** Please include screen captured images showing the mesh before and after a vertex is removed, and also include similar pairs of images to show the results of edge collapse. **You should include those images in your report** submitted as a PDF file. The images should be clearly labelled with descriptions of what is supposed to show. If the images/descriptions are unclear, you will lose some points.

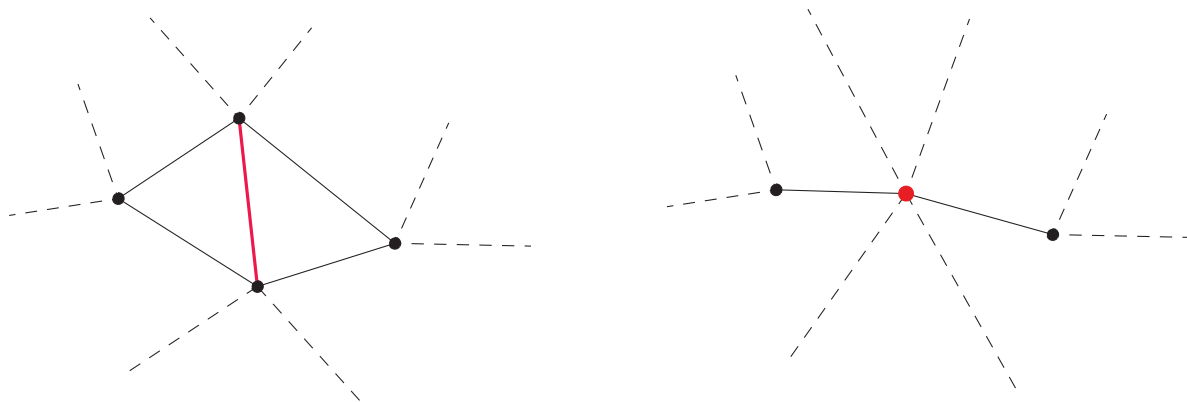


Figure 2: **Edge collapse.** Here the figures illustrate the expected change of a triangle mesh after the collapse of an edge. The **left** figure shows the triangle mesh before the edge collapse. The red edge is about to be collapsed. The **right** figure shows the resulting mesh after the red edge is removed. Note that a new vertex (shown in red) is added at the central point of the removed edge, the new vertex is connected to the vertices that originally connect to the edge, and no open boundary is produced. Please implement `HalfEdgeMesh.collapseEdge(...)` correctly to update the `HalfEdge` mesh as shown in the right figure.