CSE 5443 Lab 5a
(10 points)
Autumn, 2021

Write and submit (on carmen) an algorithm to construct a Catmull-Clark subdivision from a mesh represented by a half edge mesh data structure meshA. You should create a totally new half edge mesh data structure meshB from meshA.

Details of the half edge mesh data structure:

1. Each vertex, half edge and cell is in its own class.
2. You can extend the vertex, half edge or cell classes as needed to add extra information to each vertex, half edge and cell.
3. Each vertex has a unique integer identifier, Similarly, each half edge, and cell have unique integer identifiers. (Typically, idenfiers begin at 0. A vertex might have the same integer as a half edge or cell, and a half edge may have the same integer as a cell.)
4. A class HalfEdgeMesh stores the mesh including an array or a hashtable (depending on the implementation) of all the created vertices, half edges and cells.

Member functions (methods) of the vertex class are:

1. Dimension(): Return the dimension (number of coordinates) of the vertices. Usually 3.
2. Coord(ic): Return the ic'th coordinate.
3. Index(): Return the vertex index.
4. NumHalfEdgeFrom(): Return the number of half edges with from vertex equal to the current vertex. (Note that if the vertex is on a boundary, then this is 1 less than the number of edges incident on the vertex.)
5. KthHalfEdgeFrom(): Return the K'th half edge with from vertex equal to the current vertex. If some boundary half edge comes from the vertex, then KthHalfEdgeFrom(0) is a boundary half edge. AddCell() will make sure this condition is enforced.
6. FindHalfEdgeTo(ivto): Return half edge whose FromVertex is the current vertex and whose ToVertexIndex is ivto.

Member functions (methods) of the half edge class:

1. Index(): Return the half edge index.
2. FromVertex(): Return the vertex the half edge comes from.
3. FromVertexIndex(): Return the index of the FromVertex().
4. ToVertex(): Return the vertex the half edge goes to.
5. ToVertexIndex(): Return the index of ToVertex().
6. Cell(): Return the cell containing the half edge.
7. CellIndex(): Return the index of Cell().
8. NextHalfEdgeInCell(): Return the next half edge in the cell.
9. PrevHalfEdgeInCell(): Return the previous half edge in the cell.

10. NextHalfEdgeAroundEdge(): Return the next half edge around the edge. Equivalent to Opposite() if each edge has at most two half edges with opposite orientations.
11. PrevHalfEdgeAroundFromVertex(): Return the previous half edge around FromVertex().
12. IsBoundary(): Return true if half edge is on the mesh boundary.

Member functions (methods) of cell class:

1. Index(): Return the cell index.
2. HalfEdge(): Return a half edge in the cell.
3. NumVertices(): Return the number of cell vertices.

Member functions (methods) of HalfEdgeMesh are:

1. Vertex(iv): Return the vertex with identifier iv.
2. HalfEdge(ihalf_edge): Return the half edge with identifier ihalf_edge.
3. Cell(icell): Return the cell with identifier icell.
4. NumVertices(): Return the number of vertices in the mesh.
5. NumHalfEdge(): Return the number of half edges in the mesh.
6. NumCells(): Return the number of cells in the mesh.
7. VertexIndices(): Return a list of all the vertex indices.
8. HalfEdgeIndices(): Return a list of all the half edge indices.
9. CellIndices(): Return a list of all the cell indices.
10. MaxVertexIndex(): Return the maximum vertex index. MaxVertexIndex()+1 is an unused vertex identifier.
11. MaxHalfEdgeIndex(): Return the maximum half edge index. MaxHalfEdgeIndex()+1 is an unused half edge identifier.
12. MaxCellIndex(): Return the maximum cell index. MaxCellIndex()+1 is an unused cell identifier.
13. AddVertices(numv): Add vertices $[0, \ldots, (numv - 1)]$ to the mesh. (Assumes the mesh has no vertices.)
14. AddVertex(iv): Add vertex with index iv to the mesh.
15. AddCell(icell, Array¡int¿ cell_vertices): Add cell icell with vertices given in cell_vertices[] to the mesh. Array cell_vertices is an array of the indices of the vertices of the cell in order around the cell. Creates vertices with the given indices, if the vertices do not exist. Creates half edges as needed.
16. SetCoord(iv, coord[]): Sets the coordinates of vertex iv to coord[].