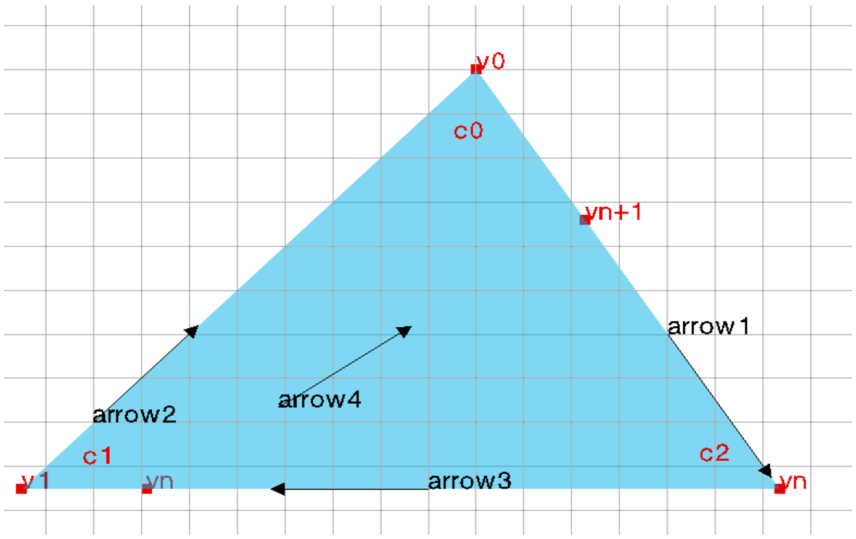
John Strauss
902455731
CS 6491
Fall 2019

Polygonal Face Graph Creation

For my face graph algorithm, I began with three arrows to define and begin testing with the first enclosed face, formed by the first the vertices (v0, v1, v2) where the lines formed by each input arrow intersect. This gives us a basis for testing further iterations of the algorithm. My next step is to subdivide the face with a new arrow and perform a new iteration in the polyloop creation. The two new vertices in the polyloop (vn, vn+1) are formed at the intersections of each previous line.



We can then calculate three new corners formed by vn and vn+1. The corner (i.e. c0, c1, c2) is defined by a face-vertex pair. After calculating each new corner, the next step is to update the corner table with each corner relation, such as c.next, c.prev, c.swing, etc. We can use the right-turn method to determine which corner is next, previous, and swing.

BSP-Tree

A BSP tree uses grouping to process data more quickly. By dividing groups into two at each iteration, it forms a tree structure. For use in calculation of the “land” or “water” state of the face graph, we will recursively iterate through every face in the graph. At each stage, with polygon P, make a node N in the BSP tree. Now, for every other polygon in the graph, add the new node to the right of node N if the polygon is land, or to the left if it is water. Next apply the recursively method to the remaining polygons in the graph. Now, the graph is organized in a way that allows us to quickly know which polygon is land and which is water.