Project 2
Group 3: Delaunay
Yu Xuan Hong & Dani Demas
02/22/2016

Project Report -
The Delaunay Triangulation Application

1. Overview

The Delaunay Triangulation Application is an interactive application that allows the user to construct a Delaunay triangulation by flipping edges of a certain triangulation on a given point set. The application automatically triangulates a point set using the incremental algorithm, and draws the circumcircles of each triangle. Using the Empty Circle Property, it shows the user the illegal edges of the triangulation by highlighting the circumcircles that contains points from the original point set. The user can choose to flip an illegal edge, and the application will return a new triangulation and corresponding circumcircles and illegal edges. Eventually, the Delaunay triangulation will be attained, where there is no illegal edge.

2. Description

See *Documentation*

3. Successes

a. Successfully built the applications as described with over 500 lines of code, implementing theorems and algorithms from references and original derivations.
b. Drafted 4-page *Documentation* that includes the detailed definitions, theorems, and algorithms used to support and build the application.
c. Created and proved multiple new algorithms and theorems (e.g. *Shortest Distance Thm.*, *Incremental Convex Hull Alg.*, etc.) that aided with the construction of desired features of application. Modified existing definitions (e.g. *Legal Edges*) and algorithms (e.g. *Incremental Triangulation Algorithm*) to facilitate implementation.
d. Application allows users to interact with point set triangulations through entering the initial point set, viewing the circumcircles and illegal edges, flipping illegal edges, receiving the updated results, and eventually attaining the Delaunay Triangulation.
e. Application is incorporates clear visualizations (coloration of circles and illegal edges) and useful functions (*Toggle* buttons), allowing user to interact with it easily.

4. **Failures**
    a. Users are unable to flip legal edges. Computationally, flipping legal edges would be more difficult than illegal ones, as the algorithm for detecting the quadrilateral containing the legal edge will likely involve point-line distance computations.
    b. Users are unable to create a triangulation on their own; instead, the application automatically triangulates the point set using the Incremental Triangulation Algorithm. To allow users to do so at their will would involve algorithms that check for the completion of a triangulation, which will likely depend on the complexity of the point set.