University of Utah
School of Computing

CS 4960 **Project #3** **Spring 2016**

Due March 1, 2016 at the start of class

Please contact the instructor Bei (beiwang@sci.utah.edu) or TA Vikram at (vikram.raj@utah.edu) for questions regarding the project. Please contact Vikram for questions regarding Blender specifically.

The project submission should include all your source code, the output files of CGAL (according to Blender format requirement described later), the input files to Blender (using the parser.c), and screen captures or outputs from Blender or some other tools that visualize your results.

For bonus points, your project submission could include the visualization of a 2D image modified by its underlying Delaunay triangulation (see details below).

For this project, you get 15 points total, with additional 5 bonus points, with a total of 20 possible points.

1 Computing and Visualizing 2D Delaunay Triangulation

Use CGAL library involving triangulation:
`http://doc.cgal.org/latest/Triangulation_2/index.html#Chapter_2D_Triangulations`
to complete the following task, you may use specifically Delaunay triangulation.

Task (15 points): Generate a set of 20+ points sampled randomly from a 2D plane, compute its Delaunay triangulation and visualize the resulting triangulation via Blender or some other visualization tools.

5 points for having a CG output file that describes the location of the hull points (in counterclockwise order). 10 points for visualizing the correct Delaunay triangulation.

2 Delaunay Triangulation Elevated

For this bonus round, you could completely ignore Task 1 stated above, but instead, using Delaunay triangulation from CGAL to create the effect found on:
`http://www.creativeapplications.net/mac/dmesh-cinder-mac/`
The coding you will need to complete for this task would include Task 1 anyway.

Task (20 points): the first 15 points include generating a set of 50+ points sampled randomly from a 2D plane where a 2D image is co-residing. Compute its Delaunay triangulation. Linear interpolate the pixel intensity values across the triangles. And finally visualize (5 points extra) the manipulated image. The visualization can be simply a pair of images: the original image (of your choice) and the modified image.