Mesh Processing
CS 3451: Project 5

1 - Objective

The topic of this project polyhedral mesh manipulation. For this project you will implement methods of creating and modifying polyhedral surfaces. For object creation, you will read a mesh description from a file. Your program should be able to display flat or smooth shaded meshes, with either white or random colors. Most importantly, your program will calculate the triangulated dual of the current object. All of the mesh objects that you create and modify will be made of only triangles. A large portion of your grade on this project will be the dual operator, so make sure you get to this!

2 - Deadline

This project should be submitted on T-Square by 11:55PM on Thursday December 7, 2017.

3 - Process

3.1 Download the base source

Download and unzip the folder with the base code for this project on T-Square.

3.2 Polyhedral File Format

The file format we will use for this project is a simple text format for meshes. The file format is basically an indexed face set. The first two lines specify the number of vertices and faces of the model, respectively. Then all of the vertex (x,y,z) values are listed, one per line. This is followed by the list of faces, one face per line. The first number that describes a face is simply the number of sides of that face, and for this project, that number will always be 3. Following this, indices into the vertex list are given. The vertices are indexed starting from the value zero. **You can open these files in any text editor**.

3.3 Project description

Your finished program will be able to read a mesh from a file and display that mesh. More importantly, with a keystroke your program will triangulated dual of the mesh, replacing the old mesh in memory. Your program should also allow toggling between flat shading (per-polygon normals) and smooth shading (per-vertex normals). Your program should obey the following keystroke commands:

1-5: Read in a mesh file (tetrahedron, octahedron, icosahedron, star, torus).

d: Run the dual operation on the current mesh (you should be able to dual more than once).

n: Toggle between per-face and per-vertex normals.

r: Assign a random color to each face of the mesh. (Set the colors once - don't make flickering colors!)

w: Change the colors of the mesh faces to white.

space: Toggle automatic rotation on and off.

3.4   Suggested Approach

First, modify the skeleton code to read in the .ply file format into your own polyhedral model data structure. Next, modify the "draw" routine to draw the current polyhedral mesh. You should then write the code needed to calculate surface normals at the vertices of a model. Modify the drawing routine so that it can toggle between per-face and per-vertex normals. Finally, write a routine that takes a given model and creates the triangulated dual of it.

Calculating the triangulated dual of a give mesh is the most challenging aspect of this project. It also counts for a significant part of your grade on this project. There are several stages to creating the dual. First, you will want to calculate the centroid of each triangle. These triangle centroids will become some of the vertices in the new dual mesh. Usually, you would then travel around a given vertex of the original mesh, connecting together these triangle centroids into a dual face. Unfortunately this would sometimes create faces with more than three edges, and we want to restrict this project to only triangles. To avoid this, you should triangulate a given dual face by calculating a new vertex that is the average of the triangle centroids. Then create the collection of new triangles that, taken together, form the dual face. Once you have created a new triangulated dual face for each vertex of the original mesh, you will have completed the triangulated dual mesh. See the octahedron and its triangulated dual in the images below to see how each original vertex is turned into a square, and how each square is in fact made of four triangles (shown by the random colors). In summary, the dual mesh is created by iterating over each vertex of the original mesh. The process of each iteration can be shown in the following 4 steps.
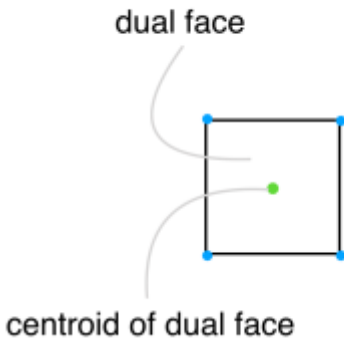
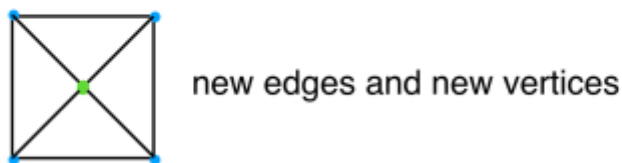### 1. Given a vertex, find the centroid on each adjacent face
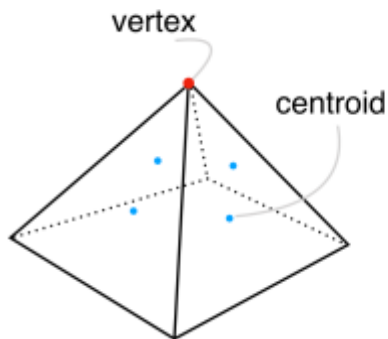
2. Connect the centroids to create a dual face

3. Find the centroid of the dual face
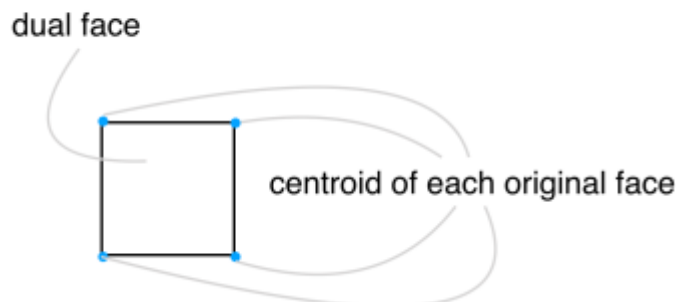
dual face

centroid of dual face

4. Connect each vertex of the dual face to the centroid of the dual face

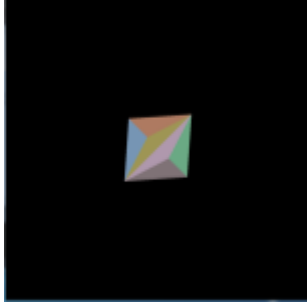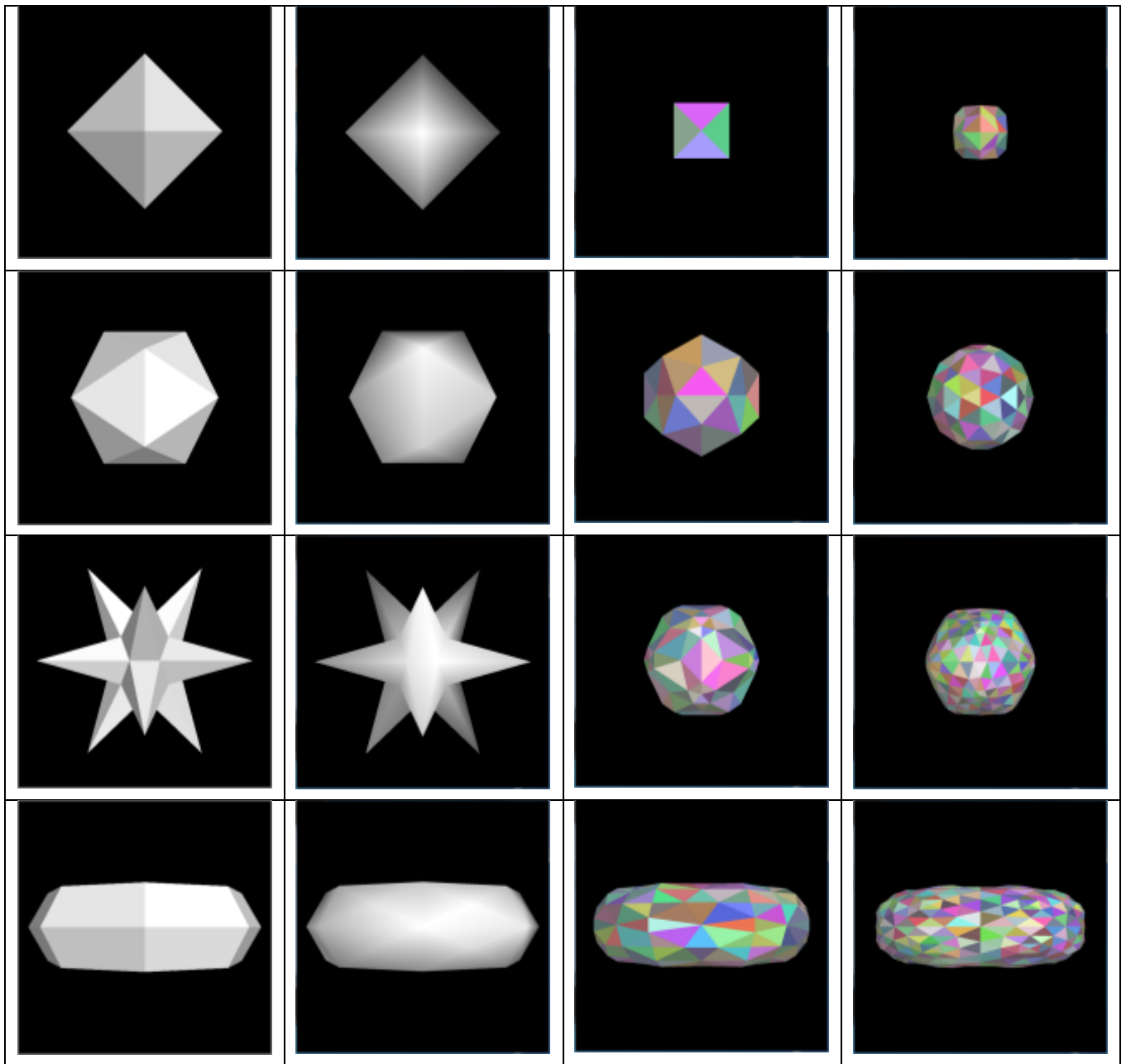new edges and new vertices

Note that you should be able to create the triangulated dual of a model more than once. Each time you calculate the dual, this will result in a mesh that has more triangles than the previous one.

We strongly recommend using the "corners" representation to store and manipulate your polygonal meshes. Note that all of the meshes for this project contain only triangles, so the corners representation is appropriate. Keep in mind that you will need mesh adjacency information to create a dual mesh and also to calculate per-vertex normals.

Sample Results

(Compare your results to those which are given. Duals are shown with 1st and 2nd keypress and with random colored faces for each shape)

Flat Shaded	Smooth Shaded	Dual (1 Iteration)	Dual (2 Iterations)

3.5   Authorship Rules

The code that you turn in entirely your own. You are allowed to talk to other members of the class and to the Professor and the TA about general implementation of the assignment. It is also fine to seek the help of others for general Processing/Java programming questions. You may not, however, use code or pseudocode that anyone other than yourself has written. Code (including pseudocode) that is explicitly not allowed includes code taken from the Web including the processing website, from books, from previous assignments or from any source other than yourself. The only exception to this is that you can make use of the example code for this assignment. You should not show your code or pseudocode to other students. Feel free to seek the help of the Professor and the TA's for suggestions about debugging your code.

3.6   Submission

In order to run the source code, it must be in a folder named after the main file. When submitting any assignment, leave it in this folder, compress it using zip and submit via T-square.