Subdivision Surfaces

CS 3451: Project 5

1 - Objective

This project is about smooth surface creation using the Loop subdivision technique. You will write a program in Processing that reads in polyhedral models and subdivides them. Your program will also be able to toggle between displaying the surface using flat and smooth shading. All of the mesh objects that you create and modify will be made of only triangles.

2 - Deadline

This project should be submitted on T-Square by 11:55PM on Tuesday April 24, 2018.

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).

s: Subdivide the current mesh (you should be able to subdivide more than once).

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

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 subdivides it using Loop subdivision. The key to this program is deciding on a polyhedral representation that will allow the subdivision and

the surface normal calculations to be done with relative ease. It would be best of you defined a mesh class that stores both the vertices and the triangles of a mesh, as well as any additional adjacency information that will help with the rest of the project. With a mesh class, you can work with two meshes at once, the original mesh and the newly created subdivided mesh.

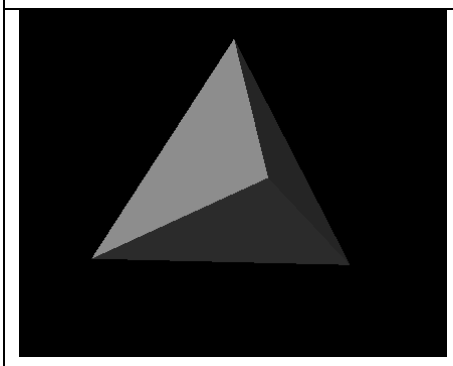
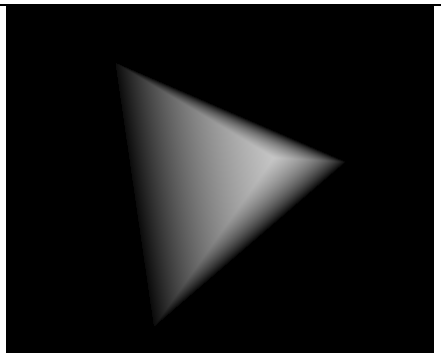
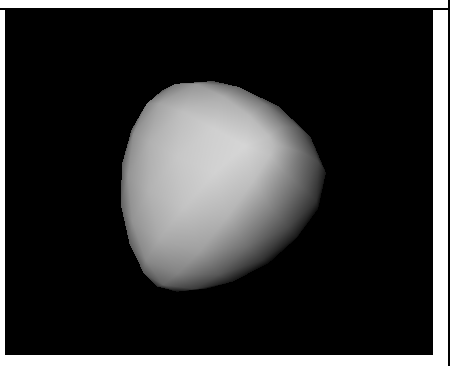
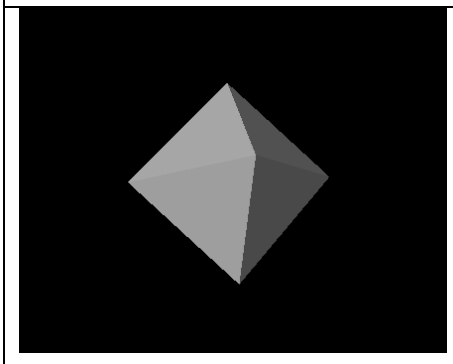
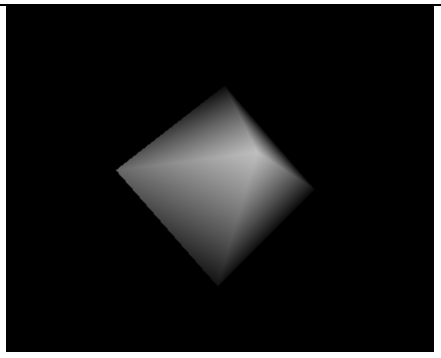
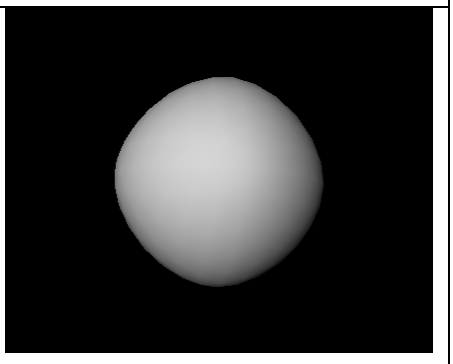
Calculating the subdivision of a give mesh is the most challenging aspect of this project. It also counts for a significant part of your grade on this project. There are several stages to creating the subdivided mesh. First, you will create new vertices that correspond to the vertices in the original mesh, but you will modify their positions according to the rules of Loop subdivision. Next, you should calculate the new vertices that are associated with each edge of the original mesh. This is the most challenging part to the assignment, because you only want to calculate **one** such vertex per edge. Note that there are two triangles that share an edge, so you need to take care not to create the vertex for a given edge twice. For this reason, it is important that you have some way of moving between the two triangles that share a given edge. Note that the **opposite** operator does this, if you use the **corners** approach to storing your triangle mesh. Finally, you should create new triangles for the new, subdivided mesh. Each original mesh triangle will be converted into four new mesh triangles.

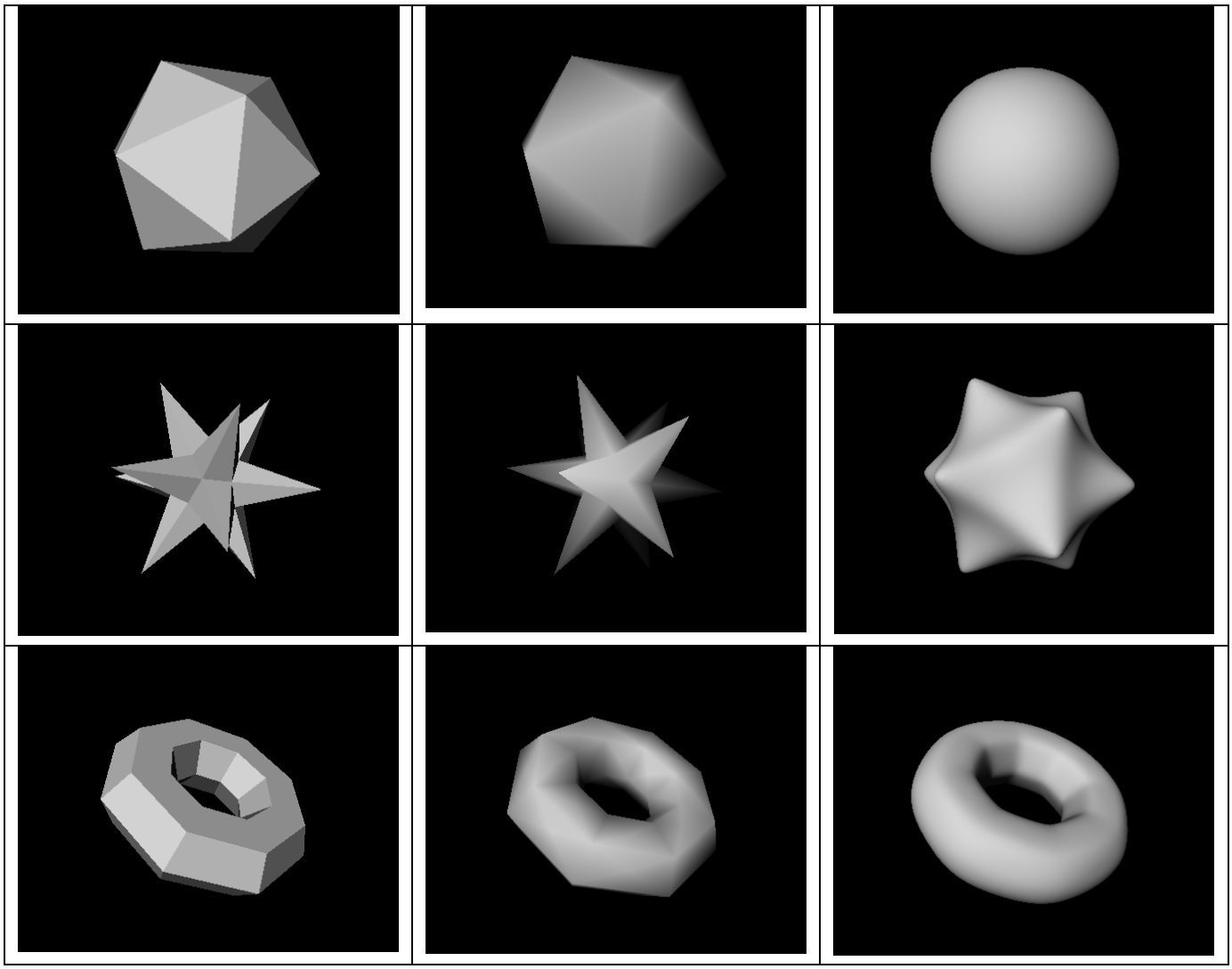
Note that you should be able to subdivide a given mesh more than once. Each time you calculate the subdivided mesh, this will result in a mesh that has more triangles than the previous one.

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 that are shown below.)

Flat Shaded	Smooth Shaded	Subdivided 3 Times
		
		



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/Python 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 Github, from other students, 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 (not rar or tar) and submit via T-square.