

CPSC 478/578 Computer Graphics
Fall 2015
Assignment #2
Assigned: Monday, September 14, 2015
Due: Monday, September 28, 2015, 11:55pm

Note that the requirements for each question may vary depending on whether you are registered for 478 or for 578. The areas addressed in this assignment are the representation of simple shapes and geometric transformations.

Turn-in Procedure

You should submit your work as a zip file using the classesv2 server. Please name your file as

LastNameFirstName-Assignment2.zip

When your file is unzipped there should be subdirectories for each question named q1, q2, etc. Name your files as directed in each question. In each directory you should have:

1. The HTML and Javascript programs you have written, or pdf's of your written response (either typed directly or scanned in). For code, you should use files in the form of the samples given, rather than producing files from scratch. This will help us follow your code.
2. If the question asks you to write code to make images, provide sample images created by your program. You can save these by clicking and saving results in your browser, or by taking a screenshot.
3. A readme.{txt, doc} that lists the input used to create the images you include. You should also list the operating system (e.g. Linux, Windows 7, 8.1, 10, Mac OS 10.4.4) and browser (e.g. Firefox 40.0.2, Safari, IEExplorer, Edge) that you used. If your programs fail on the machines used for grading, you may be asked to bring in your system to demonstrate that the files you submitted functioned in the environment you worked in.

Question I. Representing simple geometric primitives.

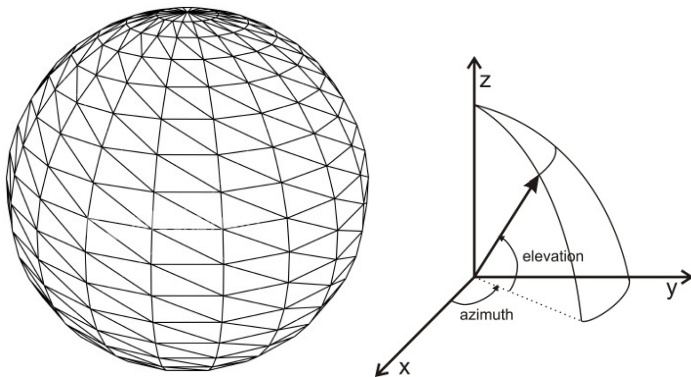
Using the primitives cube, cylinder, cone (given as code for GCPD Chapter 3) as examples, write code for additional primitives. Each should be displayed with a file primitives_*.html. You will need to alter rendering_primitives.js and primitives.js to include these. Define coordinates your new objects in the volume occupied by the cube, so that you can use the SglMat4.lookAt values used for the cube.

a) **478 and 578** A tetrahedron, a closed polygonal shape formed by 4 triangles. Your program should be tetrahedron.js

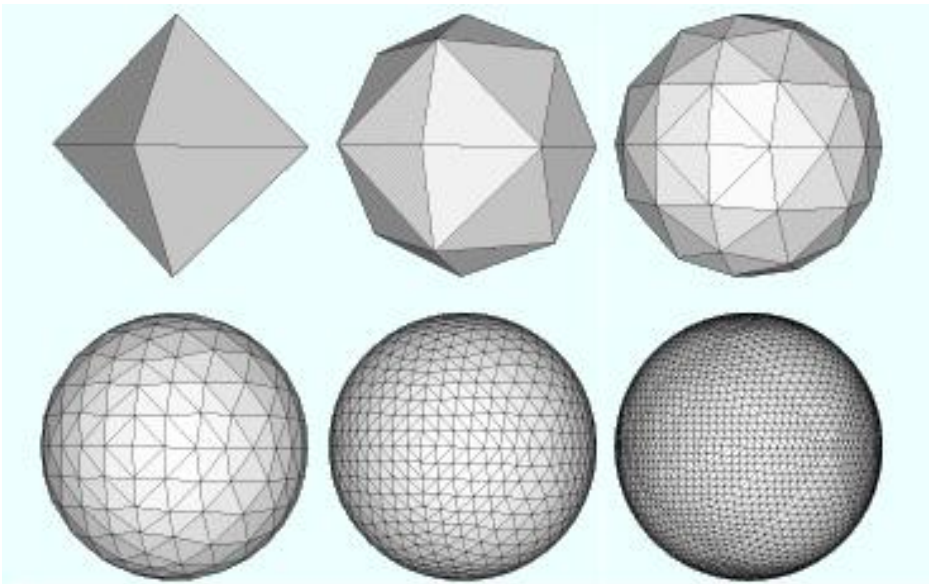
b) **478 and 578** Two ways to approximate a sphere primitive, sphere_latlong.js and sphere_subd.js

i. The sphere_latlong primitive should form triangles in a manner similar to that used for the cone and cylinder, except that you need discrete steps in both the elevation and azimuthal angles.

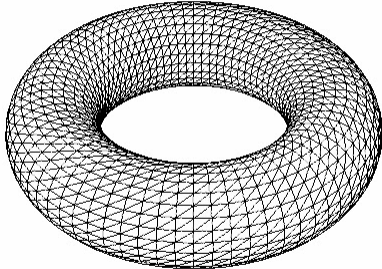
Let the number of divisions for each angle be integers specified on the html page for this primitive (user input should be in the same style as was used for input in Assignment 1).



ii. The sphere_subd primitive should start with an octahedron and successively divide each triangle face into four triangles by dividing each edge in two, and moving the midpoint to the surface of the sphere. Let the number of subdivision levels be an integer specified on the html page for this primitive.



c. **578** A program to approximate a torus. Your html page should take two integers to specify the resolution of the approximation.



Question 2. Type the answers, or scan an image of your handwritten answers.

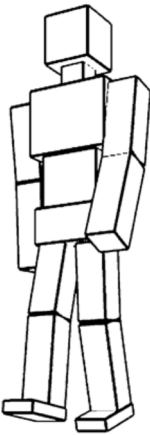
a. **478 and 578** Problem 4.13.1 from GCPD

b. **478 and 578** Problem 4.13.2 from GCPD

c. **578** Problem 4.13.3 from GCPD

d. **478 and 578** Problem 4.13.4 from GCPD

Question 3 478 and 578 Design a character from cones, cylinders, spheres and cubes that has at least 3 (**478**) or 4 (**578**) levels of hierarchy. You must use at least two (**478**) or three (**578**) different primitives (the example below only uses the cube). You must use multiple instances of each primitive. Your submission should have two parts:



a.) A tree diagram, similar to the diagrams for the table and chair example in lecture, showing primitives as child nodes, and the transformations that relate child nodes to their parent node.

b.) Create a page character.html that displays your character in the same style that the primitives were displayed.

From <http://what-when-how.com/advanced-methods-in-computer-graphics/skeletal-animation-advanced-methods-in-computer-graphics-part-1/>

An example of a character built from a hierarchy of primitives. Your character should use more types of primitives.

Question 4

478 and 578 Extend the chapter 4 2.html code so that these client related changes are implemented:

4.13.2.1 (wheel orientation) , 4.13.2.2 car transformation, 4.13.2.3 the front view.