## CSE328 Fundamentals of Computer Graphics: Programming Assignment TWO (FALL Semester, 2019)

**Programming Assignment TWO (10% of the total course grade, 400 points in total are used for the grading scheme, due at 2:20pm on WEDNESDAY, October 30)!**

**Please note that, all programming assignments/projects in this course will be graded based on (1) the correctness of program functionality, (2) the efficiency of program execution, and (3) the quality of programming codes (i.e., whether they are understandable with proper comments)!**

**This programming project focuses on applying three dimensional graphics techniques to display and edit geometric shapes in common use such as cube, tetrahedron, sphere, cylinder, cone, ellipsoid, torus, etc. Please write an OpenGL program that implements the drawing of polyhedral objects and quadrics primitives. Your program should be able to (1) display wireframe objects, (2) display flat-shaded solid objects, and (3) allow users to interactively translate and scale/zoom the displayed objects. In particular, the entire project must consist of the following components as far as the system functionalities are concerned:**

**(1) (20 points) First refer to Figure 13-1 (on Page 396 in our textbook) and read the technical contents of Chapter 13.1 and Chapter 13.2 in the book of Computer Graphics with OpenGL 4th Edition (our recommended textbook for this course). Second, display a cube object similar to the one illustrated in Figure 13-1 of our textbook.**

**(2) (20 points) Display a tetrahedron object similar to the one illustrated in Figure 13-1 (Page 396) of our required textbook.**

**(3) (60 points) (a) Carefully study the handout prepared by the instructor and fully understand its detailed technical contents (note that, the handout is a hardcopy of Pages 94-102 from the book: OpenGL Programming Guide Fifth Edition: The Official Guide to Learning OpenGL, Version 2); (b) Pay special attention to Figure 2-17 in the handout; (c) (20 points) Based on OpenGL program examples documented in the handout, write a OpenGL program to display an icosahedron of the sphere (i.e., a 20-polygon representation of the sphere); (d) (20 points) Subdivide your icosahedron once and display an improved 80-triangle approximation of the sphere (please refer to Figure 2-17); (e) (20 points) Further subdivide your 80-triangle approximation of the sphere and display a 320-triangle approximation of the sphere (i.e., subdivide your icosahedron twice).**

**(4) (60 points) Carefully study the second half of the handout (note that, it is a hardcopy of Pages 514-524 in the book: OpenGL Programming Guide Fifth Edition: The Official Guide to Learning OpenGL, Version 2) and pay special attention to the drawing instructions of spheres and cylinders as well as the OpenGL program Example 11-4 on Pages 521-524. Display a sphere, a cylinder, and a cone based on the program quadric.c shown in the handout. It may be noted that, similar examples can also be found in Chapter 13.6 (Pages 401-408) of our required textbook (Computer Graphics with OpenGL 4th Edition).**

**(5) (120 points) Based on (3) explained above, generalize your sphere drawing program to display an ellipsoid. In particular, your new ellipsoid drawing program should be capable of displaying a 20-triangle approximation, a 80-triangle approximation, and a 320-triangle approximation for the same ellipsoid, in analogy to (3) (Hint: refer to Pages 398-400 in our textbook (Computer Graphics with OpenGL 4th Edition) for the parametric equation of an ellipsoid).**

**(6) (120 points) Based on this idea (explained above) currently used to implement functionalities (1)-(5) above, draw a torus (Hint, refer to Pages 398-400 in the textbook for its parametric equation and use polygons to approximate the torus). Although you are free to select the number of polygons to be used to approximate the torus, I suggest that you sample a continuous torus using at least 25\*25=625 different points in order to better approximate a torus. Note that, the display quality should be reasonably good!**

**(7) (BONUS POINTS, up to 150 points) Based on the ideas explained above and in the class, would you be capable of extending them to the drawing of any super-quadrics. The implicit and parametric representations of any super-quadrics are available in the textbook or via the search over the internet. If you have any questions, you are strongly encouraged to discuss possible challenging issues with the instructor and his TA!!!**

**Please note that this programming assignment deadline is at the start of our lecture at 2:20pm on WEDNESDAY, October 30!**

**The project submission is done electronically. Please note that, CSE328 TA will announce the e-submission procedure in class during the next three weeks! After submitting your programming assignments/projects via e-submission, you must also write and submit a hard-copy one-page (or two-page) document explaining in details how we can execute you program and verify the above functionalities properly!**