

CG2018 Project 2: Out of 50 possible points. Due Thursday Oct 18, 2018 by 11:59pm.

Purpose: is to understand Meshes and Camera.

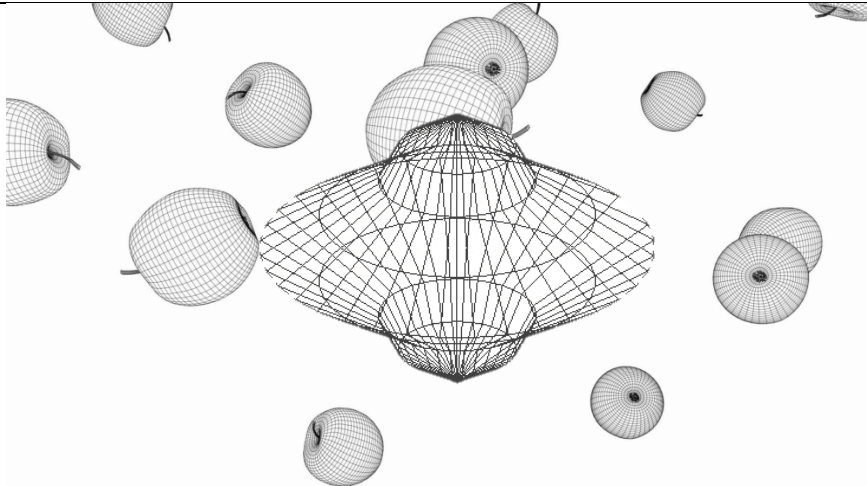
- 1) Set up 3D perspective projection in OpenGL. Set up your camera such that the camera is located at 100,100,100 looking at the origin 0,0,0.
- 2) Create a 3D scene- you choose what the scene looks like, but should be something large enough that you can “fly” through it with your camera. For each of these objects you may need to translate them so they are in view of the camera.
 - a. Create a wireframe complex 3D object using extrusion along a path or a vector; or a surface of revolution. (This object should be located at the origin, seen when the project starts) (You may use your surface of revolution you created in your homework or create a new one).
 - b. Port in a .obj file, draw a wireframe object from that .obj file. Example .obj file provided in piazza. (This object should be located near the origin, seen when the project starts)
 - c. Populate with wireframe objects from glut- these objects should be seen as the camera “flies” through the scene
- 3) Use glRotate and glTranslate to rotate/translate the camera about its own axis (not about the world coordinate system). Then add keyboard commands/callback functions to use the camera to explore the space.
 - a. #5/#1 keys = forward/back respectively along n vector
 - b. #4/#6 keys = left/right respectively along u vector
 - c. #8/#2 keys = up/down respectively along v vector
 - d. “A/a”/ “D/d” = left/ right respectively yaw
 - e. “W/w”/ “S/s” = up/down respectively pitch
 - f. “Q/q”/ “E/e” = left/right respectively roll

Notes: You will want to use the openGL commands gluLookat, glTranslate, glRotate; You may need glPush, glPop.

Additional tasks for 5450: (or extra credit for 4450).

- 4) You will be provided a Camera class. Complete the additional functions so that Roll, Pitch, and Yaw are complete. Use the camera class to “fly” through the scene. The key “C/c” should switch between the mode of camera class and OpenGL. You should create some feedback to let user know which mode it is in. (either output on command line, background color change, or text in scene to show which mode- any of those methods or other will be accepted).
- 5) Add OpenGL smooth shading to your object. You will need to make sure that you are calculating the normal properly and then drawing/rendering them with the shading.

4450: Element 1 – 5 points Element 2 – 35 points (a=10,b=15,c=10)	
--	--

<p>Element 3– 10 points</p> <p>5450</p> <p>Element 1 – 5 points</p> <p>Element 2 – 20 points (a=5,b=10,c=5)</p> <p>Element 3– 5 points</p> <p>Element 4 – 10 points</p> <p>Element 5– 10 points</p>	 <p>Simple Example, but not required to look like this.</p>
---	---

Not required, but other easy things to try:

- If the scene is flickering, you can try “GLUT_DOUBLE” for double buffering, and glutSwapBuffer command.