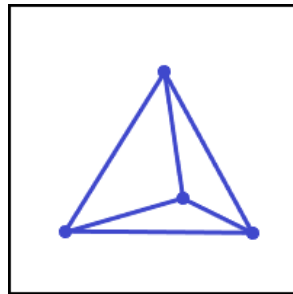

Software Assessment

Part 1:

Using 2D graphics only, create a window which will display a 3D object defined by vertices and the edges of the faces of the object. Represent the vertices using small, filled blue circles, and the edges using straight blue lines. The edges will be lines which go between the vertices, and the faces should be transparent, such that a wireframe of the 3D object is displayed. Make the object fill approximately half of the window both vertically and horizontally. Set up the coordinate frame of the window such that:

- the positive X-axis is pointing horizontally to the right,
- the positive Y-axis is pointing vertically upward,
- the positive Z-axis is pointing out of the plane of the window toward the observer, and
- the origin is at the center of the window.

Assume that the observer is an infinite distance from the canvas.



Wireframe of a simple tetrahedron

Now write code to read and display any 3D object from a comma-separated text file specified by the user. A sample object text file is attached in "object.txt." The format of the file is:

- The first line contains two integers. The first integer is the number of vertices that define the 3D object, and the second number is the number of faces that define the 3D object.
- Starting at the second line each line will define one vertex of the 3D object and will consist of an integer followed by three real numbers. The integer is the ID of the vertex and the three real numbers define the (x,y,z) coordinates of the vertex. The number of lines in this section will be equal to the first integer in the file.
- Following the vertex section will be a section defining the faces of the 3D object. The number of lines in this section will be equal to the second integer on the first line of the file. Each line in this section will consist of three integers that define a triangle that is a face of the object. The three integers each refer to the ID of a vertex from the second section of the file.

Add click and drag mouse functionality such that while the mouse button is pressed, movement of the mouse rotates the object thusly:

- Horizontal movement of the mouse rotates the 3D object about the window's Y-axis.
- Vertical movement of the mouse rotates the 3D object about the window's X-axis.
- Diagonal movement of the mouse is decomposed into vertical and horizontal components and rotates the 3D object accordingly as above.
- The point of the object nearest to the observer follows the mouse's direction

Part 2:

Create a separate program that contains all the functionality of part 1. Additionally, make each of the visible faces of the object a solid, opaque blue color. Make the color smoothly vary between #00005F (when the surface is viewed on edge, i.e. the normal of the surface makes a 90 degree angle to the Z-axis) and #0000FF (when the surface is viewed flat, i.e. orthogonal to the Z-axis) based on the angle with the Z-axis, such that the face is displayed similarly to how a shader would display it.

General Guidelines:

- Your solution should be easily executable, and well documented. All algorithms should be documented in full.
- The included sample file is an example. *Make the program work with any file in the specified format.*
- ***The intent of this assessment is to give you the opportunity to demonstrate that you can implement a basic 3D graphics display and the corresponding mathematics. Please do not use a 3D graphics package or library to do the implementation for you.***

Ways to Improve your Chances:

- Good code design & structure
- An object-oriented solution
- Good documentation!

Requirements for submission

- Turn your result in by the end of the second day. Earlier if possible.
- Please use any standard development language (e.g. C/C++, Java, Python, JavaScript or equivalent). LabView, MATLAB or equivalent are not considered to be standard development languages.
- Attach all documentation and source code in an email to cwu@neocis.com.
- For C/C++ solutions, include a working Windows executable, and for Java solutions, include an executable jar file or equivalent of your solution. JavaScript that runs in a browser or Python 3 code that executes is fine, as well, but do include any instructions needed for execution.