

Lab 14

Finally, in this lab, we render two different objects.



A simple approach is to use a different buffer for each object. Each object will need its own modelview matrix, and there will be separate calls to `gl.drawArrays` for each. We will be able to use the same shaders.

This lab is an extension of Lab 12. The vertex shader will be pass through, and not have the matrix definitions we used in Lab 13.

The buffer variables will be global, since they will be declared in `init()` and then used in `render()`.

Initially, the code will render a color cube, translated to the left.

You'll modify the code in two places, following the example of the color cube.

After the color cube definitions:

- (1) – Define four vertices for the tetrahedron in `verticesTetra`
- (2) – Complete function `triangle` to push the vertices and color for a triangle face
- (3) – Complete function `colorTetra` to define the vertices (and color) of the four faces, by calling `triangle` four times.

In `render()`:

- (4) – Set up the color buffer for your tetrahedron
- (5) – Bind and send vertex info for your tetrahedron to the vertex shader
- (6) – Uncomment the code to draw the four triangles of the tetrahedron