***Lab 21***

We will modify the code in `Lab21ShadowMap.js`, which is based on `08/ShadowMap.js`.

1 – First, just run the code. Do you get a "*`FrameBuffer not complete`*" message after several seconds? If so, you can modify the call to render so it is in a timer, giving the page a chance to render before the buffers swap: `setInterval( requestAnimationFrame(render) , 500) ;` Does this clear up the problem?

2 – Next, modify the code so that the shadow is projected onto a plane, and not a cube.
It's probably easiest to pick an appropriate face of the cube (one that is mostly opposite the triangle casting the shadow) and to just render that one face of the cube.
To do this, you can:
a – just remove all but one call to quad() in colorCube(), keeping the face you want (try using vertices 3, 0, 4, and 7).
b – in the ***two*** render passes, a total of ***four*** calls to `gl.drawArrays`. change the parameters giving the length of the primitives There are only 6 vertices needed for the plane now, instead of 36 for the cube.

3 – Once your code is projecting a shadow onto a plane, clean it up by removing unused vertex definitions and using ***plane*** to name objects, instead of ***cube***.

4 – Finally, do you notice aliasing, especially on the shadow on the plane? While shadow maps are fairly easy to implement, aliasing can still be a problem, because discrete data is being projected between two different coordinate systems. This is even with a texture map with a resolution of 1024 x 1024.