Lab 17

First, modify the `Lab17TextureSettings.js` to render a triangle. You'll need to define `vertexColor` and three (x,y) points in the `points` array, in clip coordinates. Start with a simple 2D isoceles triangle.

Next, add the checkerboard texture map. Look at `LabKey/Lab17TextureSettingsKEY` and compare the code. For your convenience, the lines of code to add are provided here:

In `Lab17TextureSettings.html`:

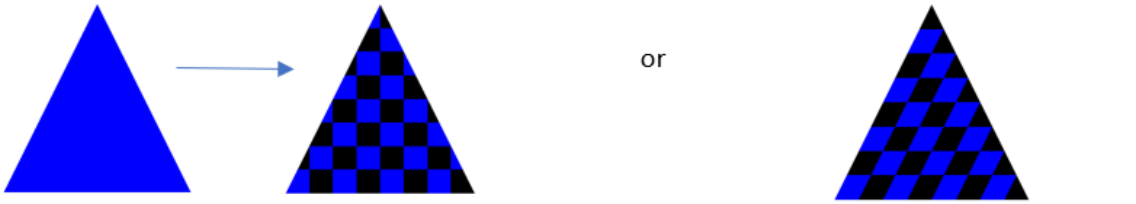
```
in vec2 aTexCoord;
out vec2 vTexCoord;
  vTexCoord = aTexCoord;
in vec2 vTexCoord;
  // initial, masked, underlying color will show
  fColor = vColor*texture(uTextureMap, vTexCoord);
  // not masked, will overwrite underlying colored triangle
  //fColor = texture(uTextureMap, vTexCoord);
```

In `Lab17TextureSettings.js`:

```
var texSize = 64;
// Create a checkerboard pattern using floats
var image1 = new Array()
    for (var i =0; i<texSize; i++)  image1[i] = new Array();
    for (var i =0; i<texSize; i++)
        for ( var j = 0; j < texSize; j++)
           image1[i][j] = new Float32Array(4);
    for (var i =0; i<texSize; i++) for (var j=0; j<texSize; j++) {
        var c = (((i & 0x8) == 0) ^ ((j & 0x8) == 0));
        image1[i][j] = [c, c, c, 1];
    }

// Convert floats to ubytes for texture
var image2 = new Uint8Array(4*texSize*texSize);
    for (var i = 0; i < texSize; i++)
        for (var j = 0; j < texSize; j++)
           for(var k =0; k<4; k++)
                image2[4*texSize*i+4*j+k] = 255*image1[i][j][k];


var texCoordsArray = [];

var texCoord = [
    vec2(0, 0),
    vec2(0, 1),
    vec2(1, 1)
];

function configureTexture(image) {
    var texture = gl.createTexture();
    gl.activeTexture(gl.TEXTURE0);
    gl.bindTexture(gl.TEXTURE_2D, texture);
    gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, texSize, texSize, 0,
gl.RGBA, gl.UNSIGNED_BYTE, image);
    gl.generateMipmap(gl.TEXTURE_2D);
```

```
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER,
gl.NEAREST_MIPMAP_LINEAR);
//        gl.NEAREST_MIPMAP_NEAREST);
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MAG_FILTER,
gl.NEAREST);
//    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MAG_FILTER,
gl.NEAREST_MIPMAP_LINEAR);
}

    texCoordsArray.push(texCoord[0]);
    texCoordsArray.push(texCoord[1]);
    texCoordsArray.push(texCoord[2]);


    // texture buffer
    var tBuffer = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, tBuffer);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(texCoordsArray),
gl.STATIC_DRAW);
    var texCoordLoc = gl.getAttribLocation(program, "aTexCoord");
    gl.vertexAttribPointer(texCoordLoc, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(texCoordLoc);
    configureTexture(image2);
    gl.uniform1i( gl.getUniformLocation(program, "uTextureMap"), 0);
```

You can try try different texture vertices: and different settings in glTexParameteri,



no texture

texture vertices

(0.5,1) , (0,0) , (1,0)

texture vertices

(1,1) , (0,0) , (1,0)

and different settings in `glTexParameteri`, check `LabKey/Lab17TextureSettingsKEY.js` for other examples.