Question 1

a)

Uniform A global variable that cannot be changed by the shaders, and is dependent upon the geometric primitive in use.

Attribute A global variable that cannot be changed by the shader, which is dependent upon the current vertex, and therefore is only available within the vertex shader.

Varying A variable that stores data that has been interpolated, so can vary pixel by pixel. The vertex shader is able to modify this, however the fragment shader is not.

b)

```
var polygonVerticesAndColurs = new Float32Array([
     0.0,  1.0,  -4.0,  0.4,  1.0,  0.5,
    -0.5, -1.0,  -4.0,  0.4,  0.7,  0.4,
     0.5, -1.0,  -4.0,  0.5,  0.4,  0.5,
     0.0,  1.0,  -2.0,  1.0,  1.0,  0.4,
    -0.5, -1.0,  -2.0,  0.3,  1.0,  0.5,
     0.5, -1.0,  -2.0,  1.0,  0.4,  0.4
]);
var n = 6; // 6 vertices in this polygon

// Create a buffer object
var vertexColorbuffer = gl.createBuffer();
if (!vertexColorbuffer) {
  console.log('Failed to create the buffer object');
  return -1;
}

// Write the vertex information and enable it
gl.bindBuffer(gl.ARRAY_BUFFER, vertexColorbuffer);
gl.bufferData(gl.ARRAY_BUFFER, polygonVerticesAndColurs, gl.STATIC_DRAW);
```

Below is further code relating to using this array, as I was unsure how much code needed to be included.

```
var FSIZE = verticesColors.BYTES_PER_ELEMENT;

// Assign the buffer object to a_Position and enable the assignment
var a_Position = gl.getAttribLocation(gl.program, 'a_Position');
if(a_Position < 0) {
  console.log('Failed to get the storage location of a_Position');
  return -1;
}

gl.vertexAttribPointer(a_Position, 3, gl.FLOAT, false, FSIZE * 6, 0);
gl.enableVertexAttribArray(a_Position);

// Assign the buffer object to a_Color and enable the assignment
var a_Color = gl.getAttribLocation(gl.program, 'a_Color');
if(a_Color < 0) {
  console.log('Failed to get the storage location of a_Color');
  return -1;
}
gl.vertexAttribPointer(a_Color, 3, gl.FLOAT, false, FSIZE * 6, FSIZE * 3);
gl.enableVertexAttribArray(a_Color);
```

c)

TODO SCENE GRAPH

d) TODO

i.

ii.

Question 2

The included files `tqvj24.html` and `tqvj24.js` are my code for this question.