Web GL Fundamentals - Manipulation and Image Processing

IF3260 Grafika Komputer

Ryo Richardo 13519193

1. How it works

Pada bagian ini, Web GL Fundamentals menjelaskan tentang sintaks dasar untuk membentuk objek 2 dimensi sederhana, yaitu segitiga. Untuk melakukan hal tersebut, dibuat 2 fungsi untuk membantu mendefinisikan sebuah segitiga sebagai berikut:

|  |
| --- |
| // Fill the buffer with the values that define a triangle.  function setGeometry(gl) {  gl.bufferData(  gl.ARRAY\_BUFFER,  new Float32Array([  0, -100,  150, 125,  -175, 100]),  gl.STATIC\_DRAW);  } |
| // Draw the scene.  function drawScene() {  ...  // Draw the geometry.  var primitiveType = gl.TRIANGLES;  var offset = 0;  var count = 3;  gl.drawArrays(primitiveType, offset, count);  } |

Perlu diperhatikan pada fungsi setGeometry, didefinisikan 3 buah koordinat dalam sebuah buffer yang mewakilkan posisi dari setiap titik segitiga yang akan digambar. Selain itu, didefinisikan juga pada fungsi drawScene bentuk segitiga dalam pemanggilan konstanta gl.TRIANGLES dan deklarasi jumlah sudut pada variabel count = 3.

Latihan berikutnya adalah membentuk lebih dari 1 segitiga dan memberikan warna yang konstan. Dilakukan perubahan untuk fungsi setGeometry, drawScene, dan penambahan fungsi baru setColors sebagai berikut:

|  |
| --- |
| // Fill the buffer with the values that define a triangle.  function setGeometry(gl) {  gl.bufferData(  gl.ARRAY\_BUFFER,  new Float32Array([  -150, -100,  150, -100,  -150, 100,  150, -100,  -150, 100,  150, 100]),  gl.STATIC\_DRAW);  } |
| // Draw the scene.  function drawScene() {  ...  // Draw the geometry.  var primitiveType = gl.TRIANGLES;  var offset = 0;  var count = 6;  gl.drawArrays(primitiveType, offset, count);  } |
| // Fill the buffer with colors for the 2 triangles  // that make the rectangle.  function setColors(gl) {  // Pick 2 random colors.  var r1 = Math.random();  var b1 = Math.random();  var g1 = Math.random();    var r2 = Math.random();  var b2 = Math.random();  var g2 = Math.random();    gl.bufferData(  gl.ARRAY\_BUFFER,  new Float32Array(  [ r1, b1, g1, 1,  r1, b1, g1, 1,  r1, b1, g1, 1,  r2, b2, g2, 1,  r2, b2, g2, 1,  r2, b2, g2, 1]),  gl.STATIC\_DRAW);  } |

Perhatikan bahwa pada kasus ini, terdapat 6 koordinat titik pada buffer untuk mewakilkan keenam titik dari 2 segitiga yang dibentuk. Jumlah sudut segitiga juga berubah menjadi 6 yang didefinisikan pada variabel count = 6 pada fungsi drawScene. Untuk menambahkan warna, ditambahkan pula buffer yang berisi warna pada setiap sudut segitiga dalam format RGBA pada fungsi setColors. Semua fungsi tersebut nantinya akan dipanggil dalam program utama dan kemudian di-bind dengan objek canvas.

Latihan yang terakhir adalah membuat gradasi warna pada objek. Caranya adalah dengan menambahkan nilai yang berbeda di setiap titik dari sudut-sudut objek 2 dimensi yang didefinisikan sebelumnya. Untuk merealisasikannya, dilakukan perubahan terhadap fungsi setColors sebagai berikut:

|  |
| --- |
| // Fill the buffer with colors for the 2 triangles  // that make the rectangle.  function setColors(gl) {  // Make every vertex a different color.  gl.bufferData(  gl.ARRAY\_BUFFER,  new Float32Array(  [ Math.random(), Math.random(), Math.random(), 1,  Math.random(), Math.random(), Math.random(), 1,  Math.random(), Math.random(), Math.random(), 1,  Math.random(), Math.random(), Math.random(), 1,  Math.random(), Math.random(), Math.random(), 1,  Math.random(), Math.random(), Math.random(), 1]),  gl.STATIC\_DRAW);  } |

Perlu diperhatikan bahwa pada kasus ini, setiap sudut diberikan nilai RGBA yang unik secara random, sehingga terjadi perbedaan warna di sudut-sudut segitiga. Hal ini membuat gradasi warna secara otomatis pada area dalam segitiga. Jika sudah mencapai tahap ini, maka hasil program akan tampak seperti gambar berikut:

A picture containing application

Description automatically generated