Tessellator's Delight

Introduction to WebGL for OpenStreetMap
State of the Map EU 2014

Brett Camper
Mapzen

WebGL

Power of your GPU, exposed in JavaScript

New mapping aesthetics & data visualization

But…

WebGL

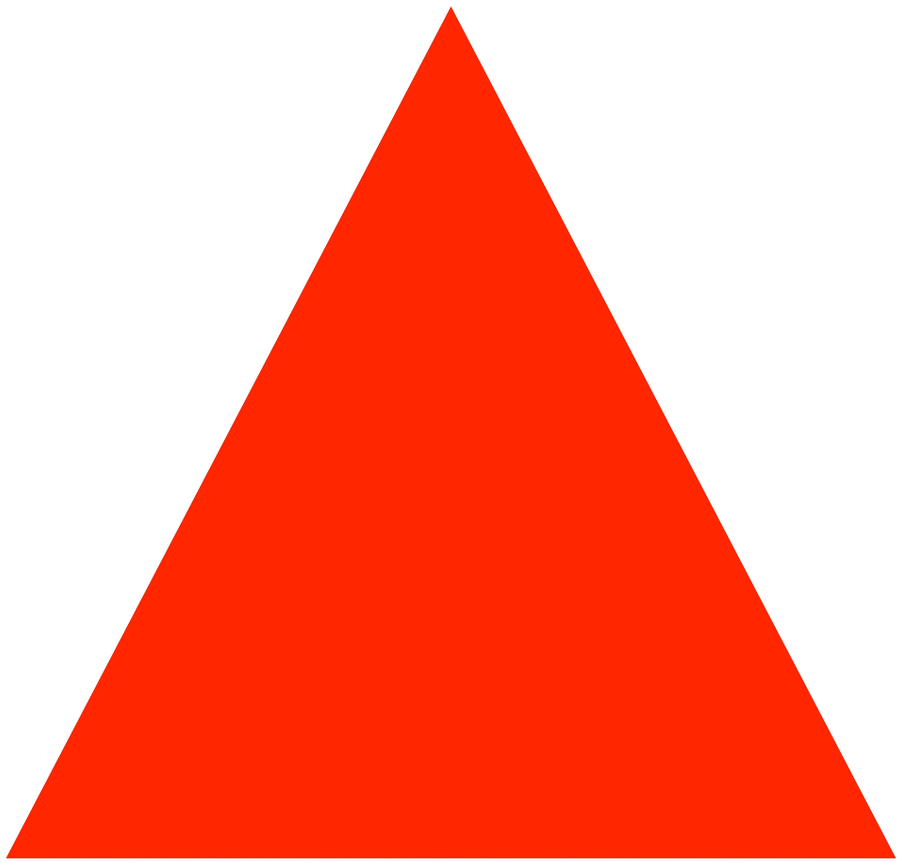
Power of your GPU, exposed in JavaScript

New mapping aesthetics & data visualization

But…

Not enough people doing it!

From this…

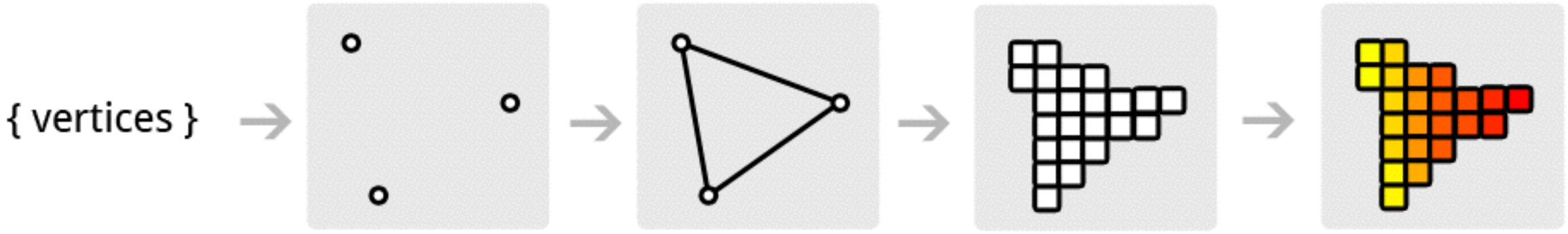


…to this

Part 1: the triangle

How a triangle is born: the WebGL pipeline

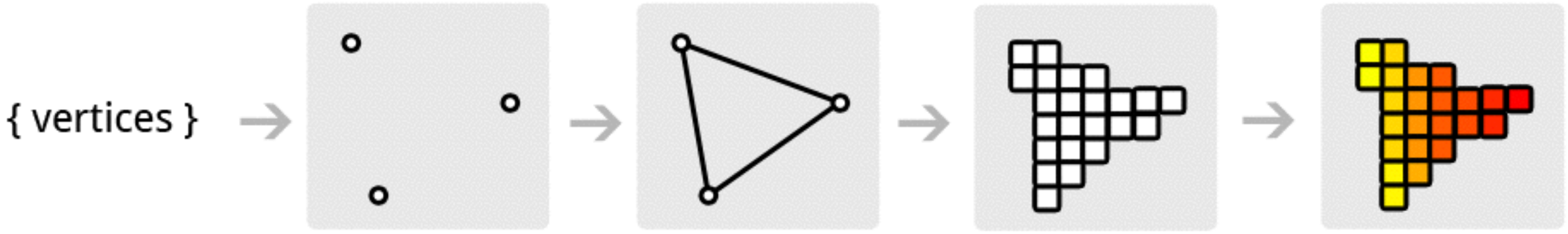
Vertex Data **Vertex Shader** **Shape Assembly** **Rasterization** **Fragment Shader**



credit: http://open.gl/

How a triangle is born: the WebGL pipeline

Vertex Data **Vertex Shader** **Shape Assembly** **Rasterization** **Fragment Shader**

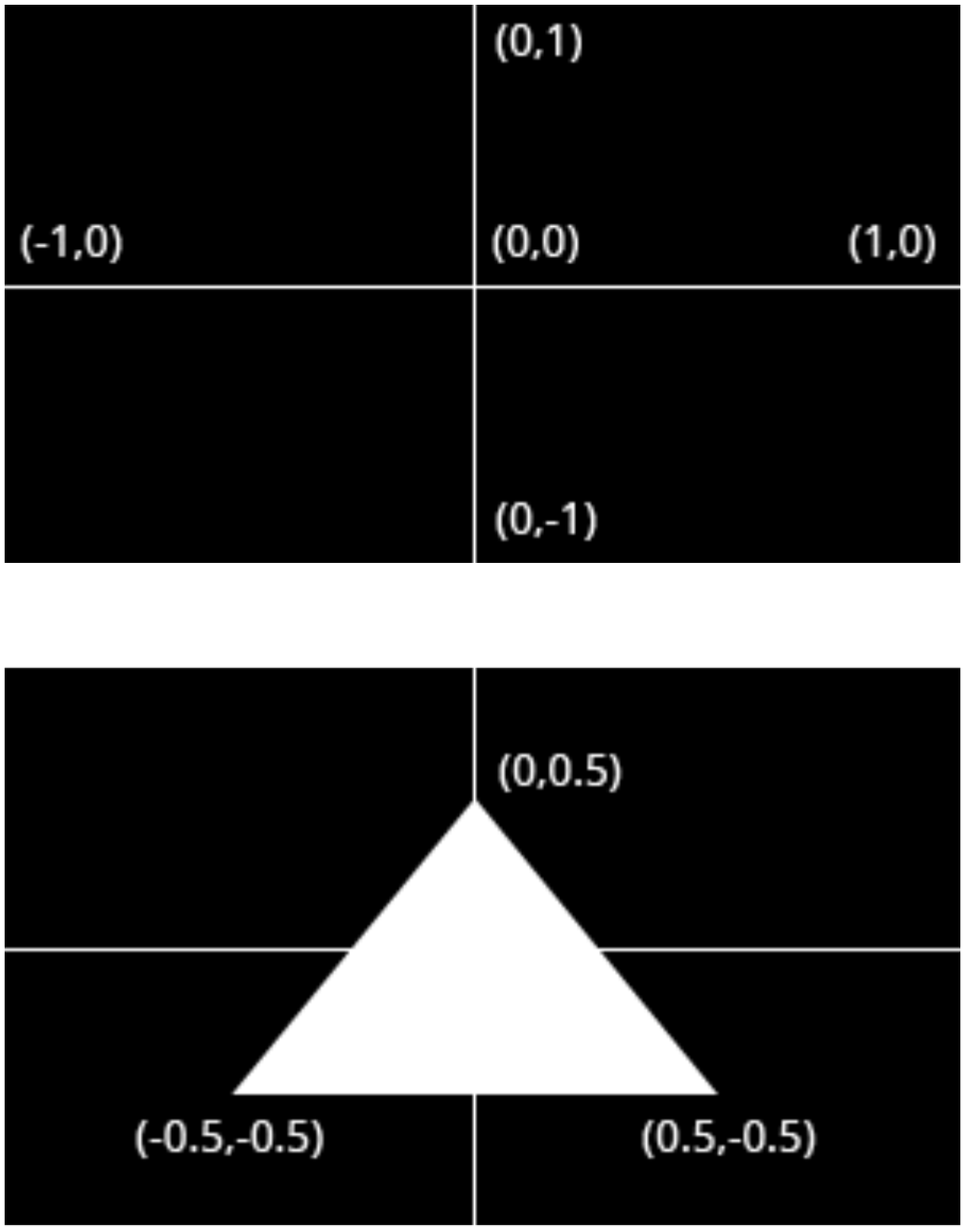


credit: http://open.gl/

Vertex Data: a Triangle

credit: http://open.gl/

Vertex Data: a Triangle



credit: http://open.gl/

Vertex Data: a Triangle

```
var vertices = new Float32Array([
    0.0,  0.5,    // Vertex 1 (x, y)
    0.5, -0.5,    // Vertex 2 (x, y)
   -0.5, -0.5     // Vertex 3 (x, y)
]);
```

Shaders?!

Small programs run on the GPU

Written in GLSL (GL shading language)

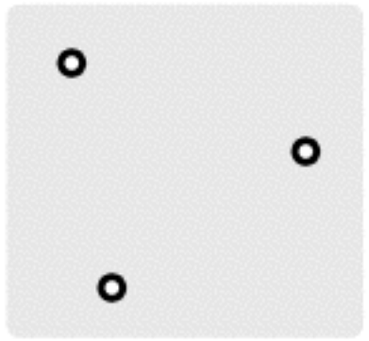
Manipulate geometry & pixel data as they are drawn

Highly parallelized & compartmentalized

Vertex Shader

Vertex Data **Vertex Shader**

{ vertices }

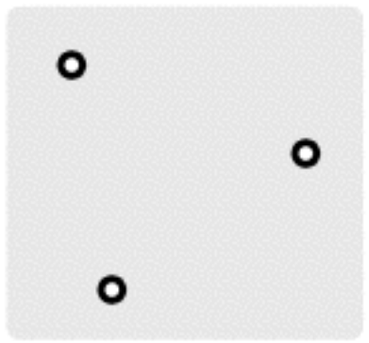


Vertex Shader

Called for each **vertex** in a triangle

Vertex Data **Vertex Shader**

{ vertices }



Vertex Shader

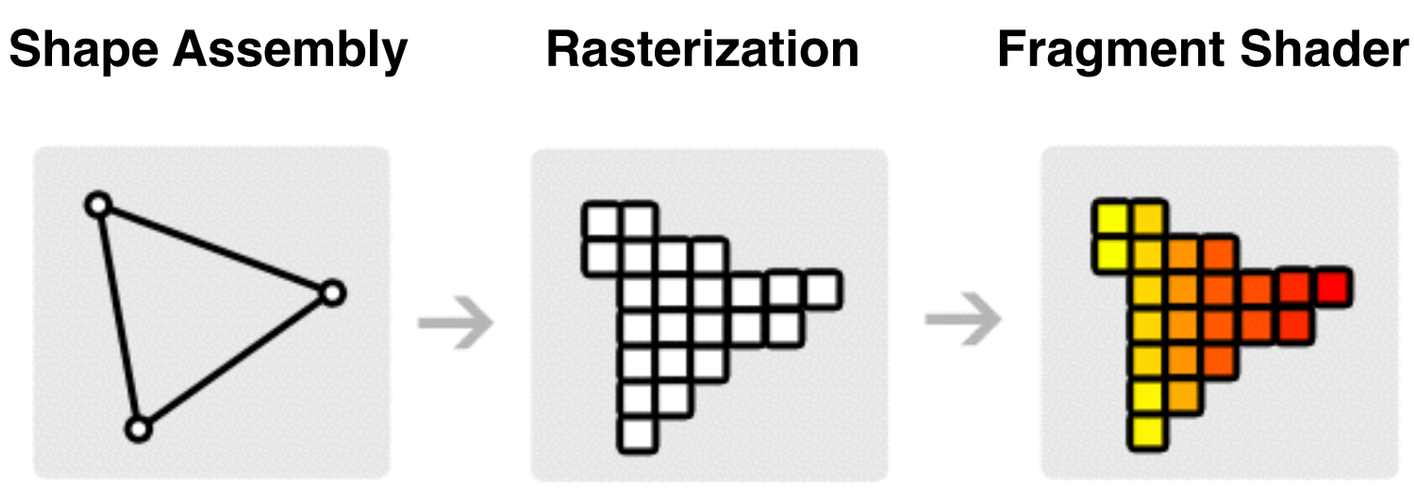
Where on screen should this vertex be drawn?

Vertex Shader

Where on screen should this vertex be drawn?

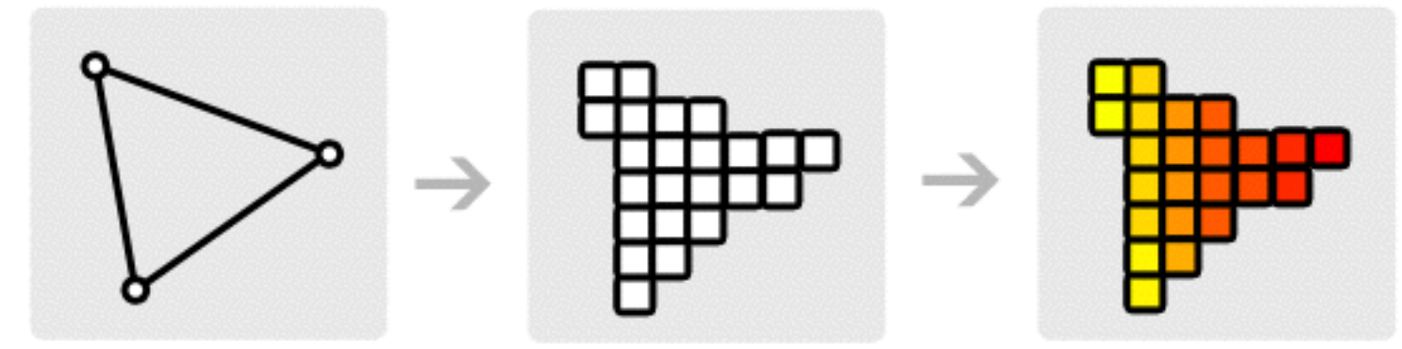
```
// Input: vertex position from data buffer
attribute vec2 position;

void main () {
    // Output: vertex position in clipspace coordinates, [-1, 1]
    gl_Position = vec4(position.x, position.y, 0.0, 1.0);
}
```

Fragment Shader



Fragment Shader

Called for each **pixel** in a triangle



Fragment Shader

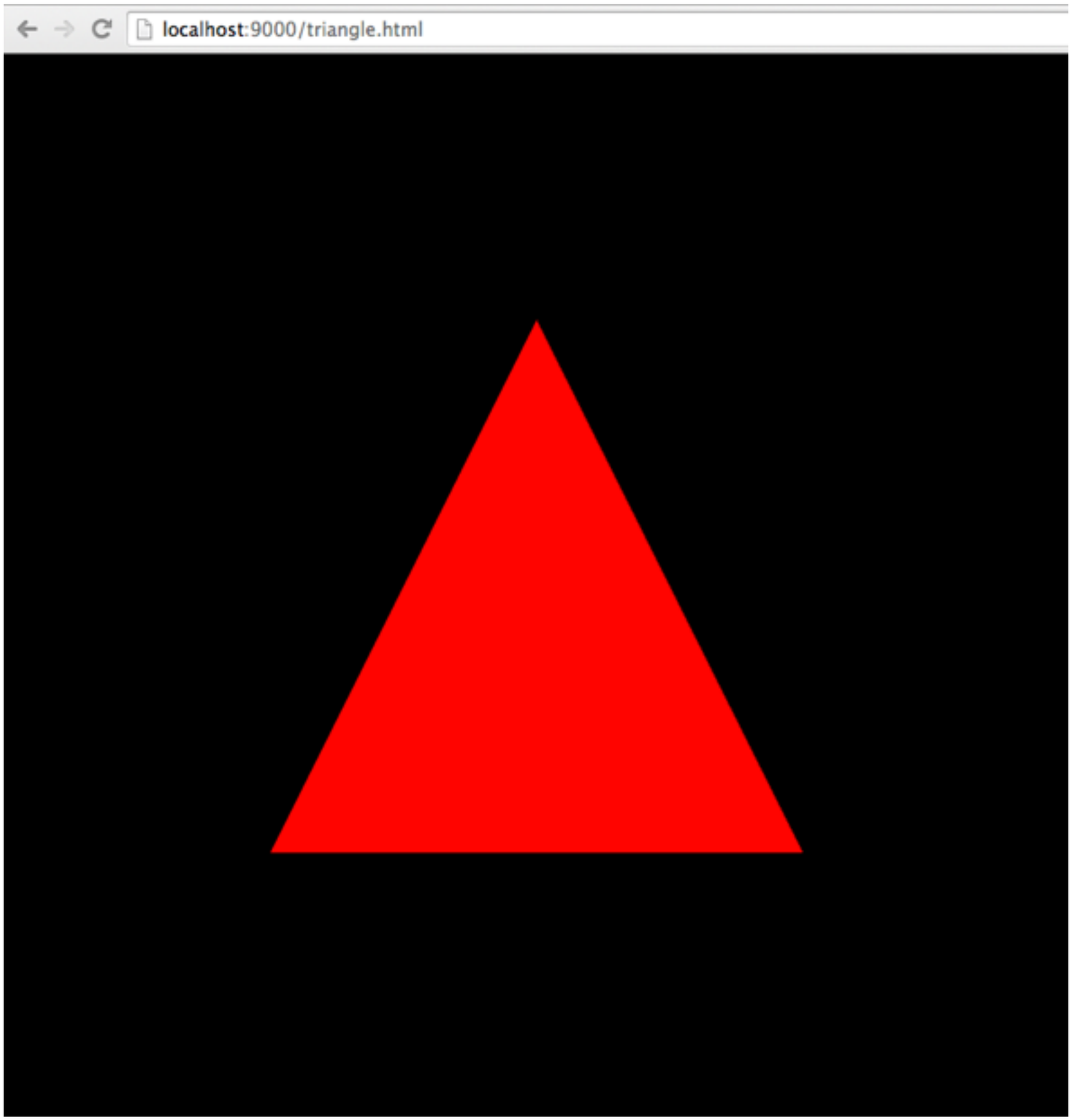
What color should this pixel be?

Fragment Shader

What color should this pixel be?

```
void main () {
      // Each pixel is a 4-component RGBA value.
      // Set every pixel to red:
      gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
}
```

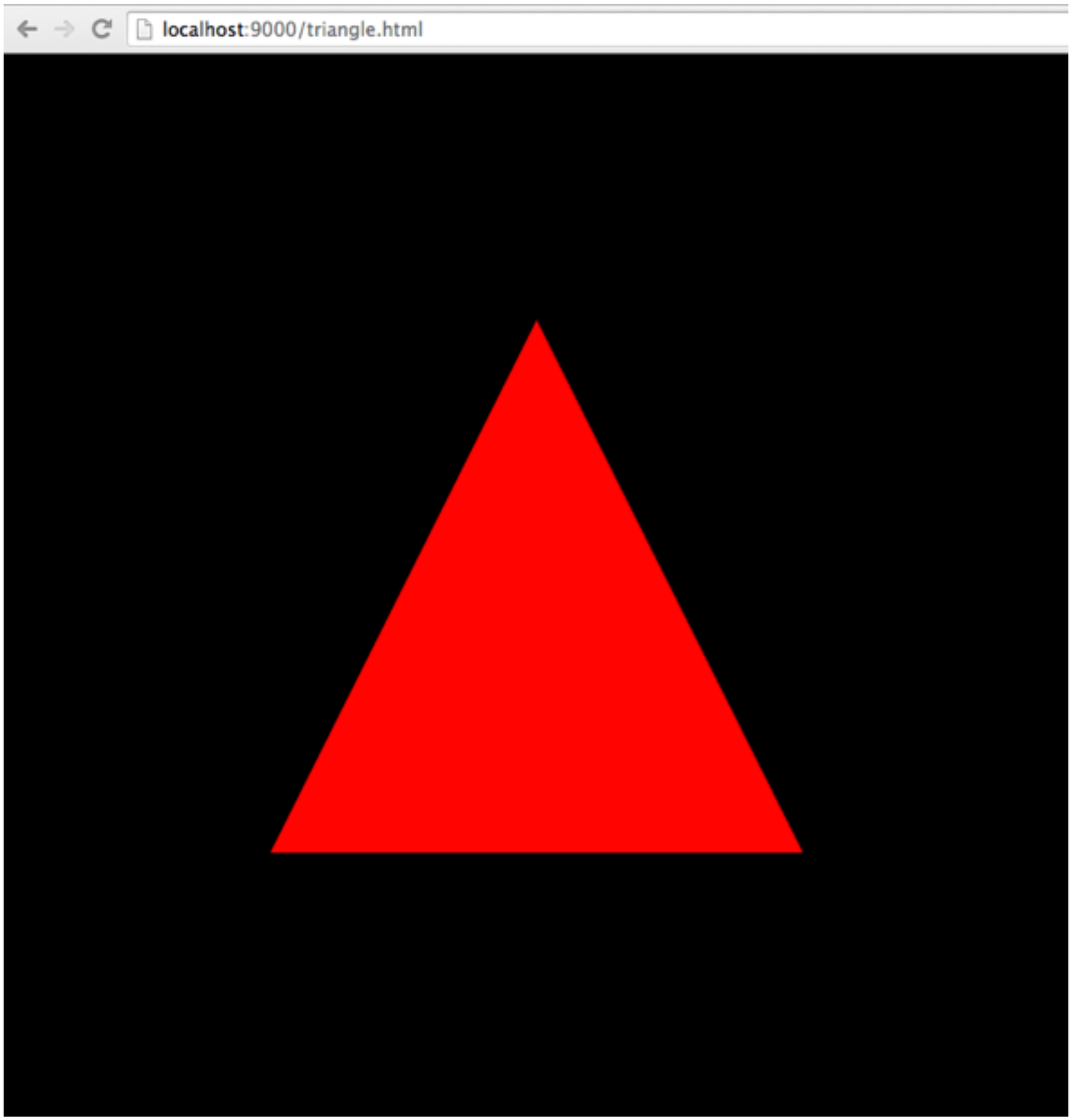
Triangle!



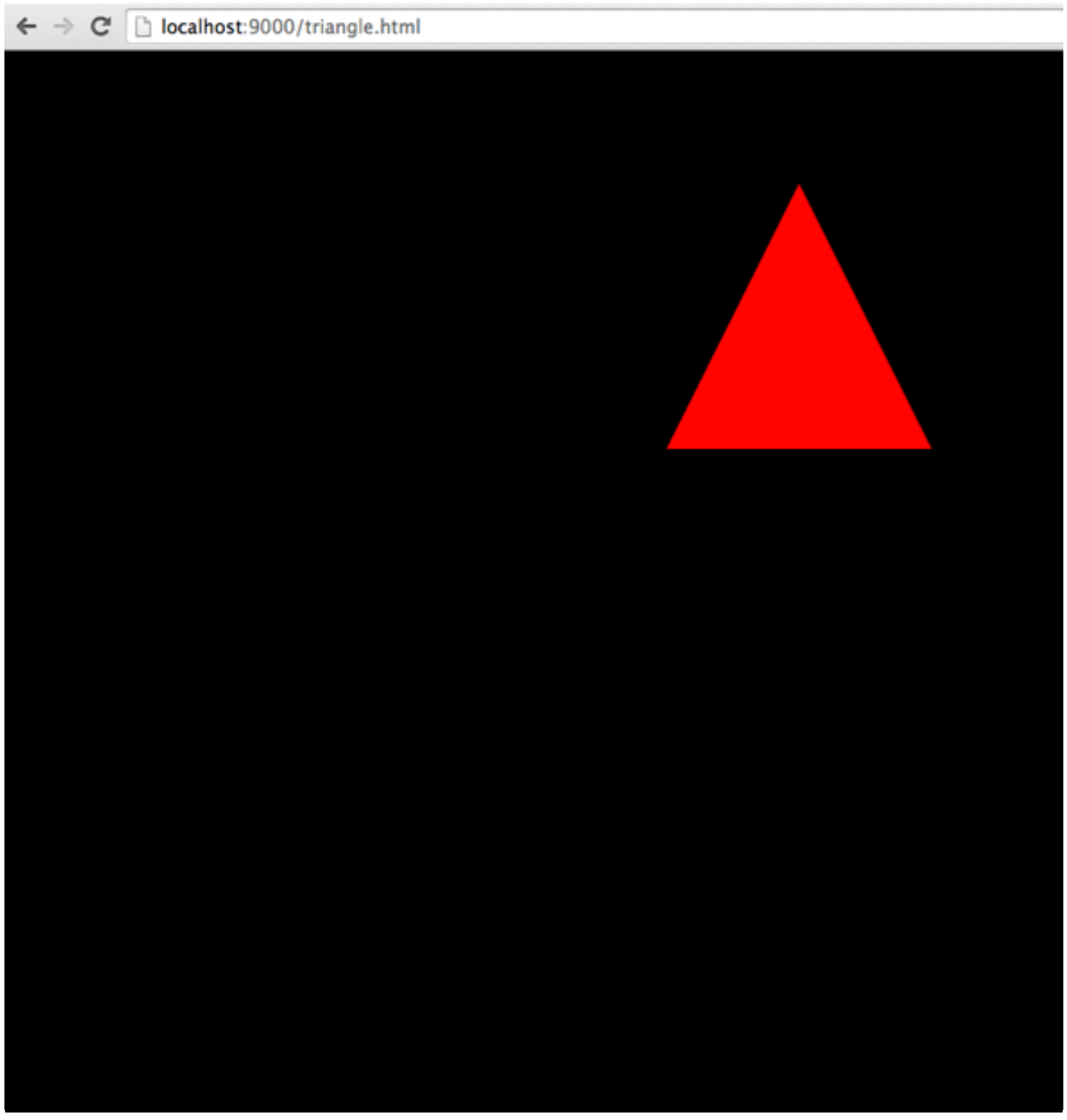
Vertex Shader: Transforms

```
// Input: vertex position from data buffer
attribute vec2 position;

void main () {
    // Copy the vertex to a new variable and transform it
    vec2 p = position;
    p *= 0.5;     // scale vertex down
    p.x += 0.5;    // move vertex right
    p.y += 0.5;    // move vertex up

    // Vertex position in clipspace coordinates, [-1, 1]
    gl_Position = vec4(p.x, p.y, 0.0, 1.0);
}
```

Vertex Transformed Triangle!



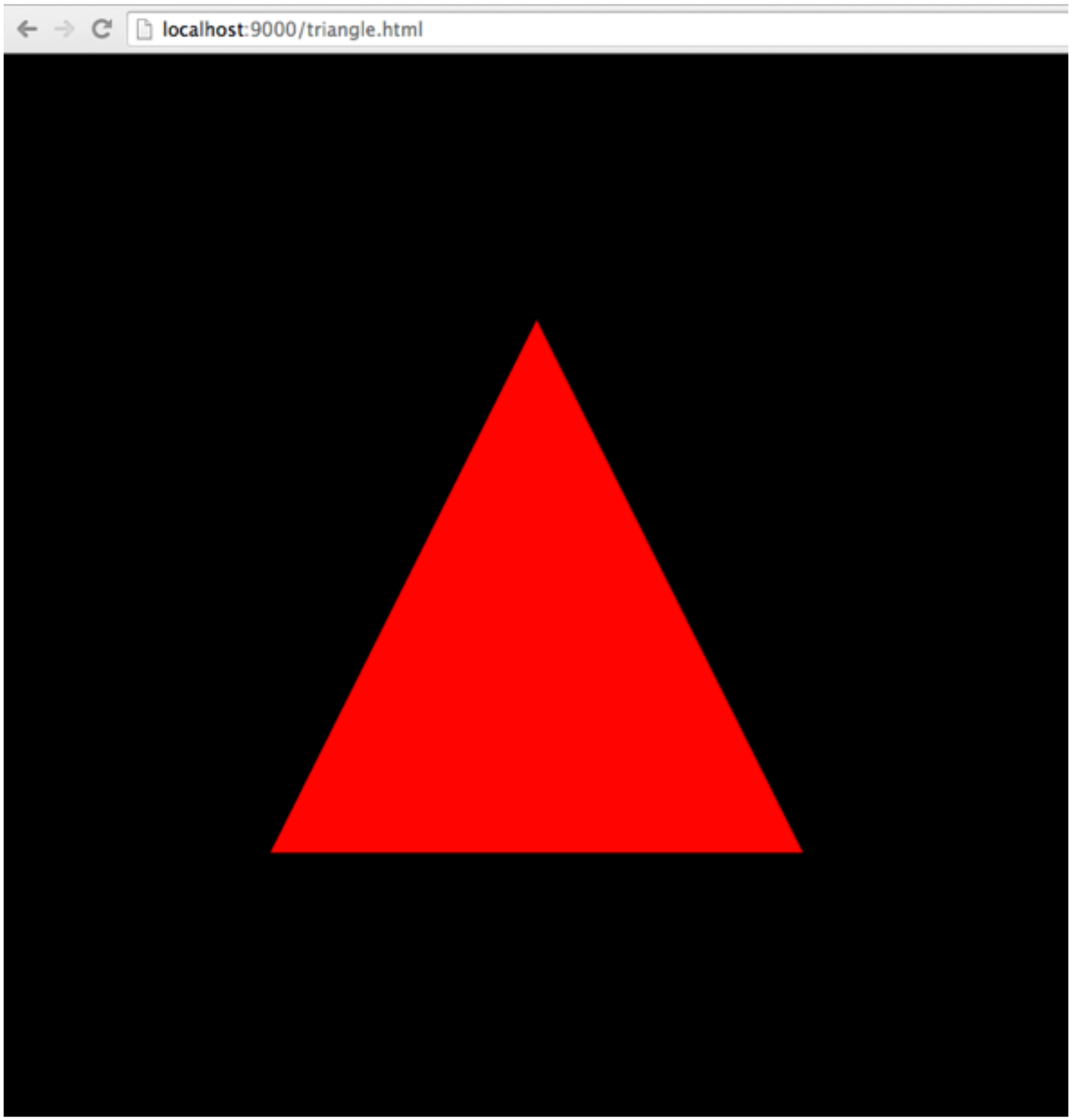
Vertex Transformed Triangle!



Fragment Shader: Transforms

```
void main () {
      // Each pixel is a 4-component RGBA value.
      // Vary color by position:
      gl_FragColor = vec4(
         gl_FragCoord.x / 1000.0,
         gl_FragCoord.y / 1000.0,
         0.0,
         1.0);
   }
```

Fragment Transformed Triangle!



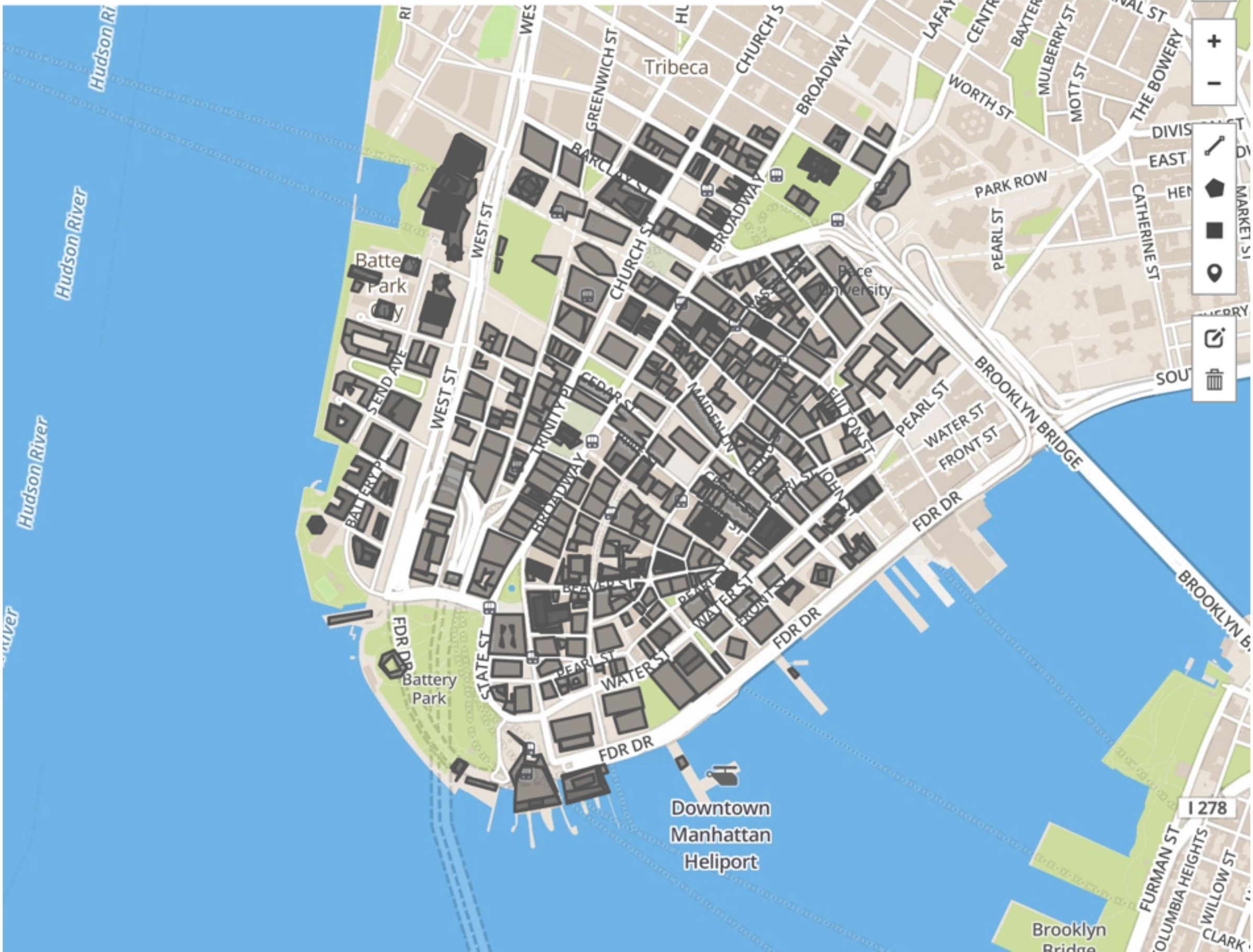
Fragment Transformed Triangle!

Part 2: the map

Vector tiles!

```
{
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          -74.017923,
          40.711239
        ],
        [
          -74.017853,
          40.711423
        ],
        [
          -74.017079,
          40.711252
        ],
        [
          -74.017149,
          40.711069
        ],
        [
          -74.017923,
          40.711239
        ]
      ]
    ]
  },
  "type": "Feature",
  "id": "76abbd6eea",
  "properties": {
    "name": "Gateway Plaza 400",
    "height": 88
  }
}
```

Open Save New Share anonymous/979fd38d3be2f616d305



Mapzen Vector Tile Service

http://mapzen.com/vector

Easy access to OSM data
Worldwide coverage, updated daily
Geometry clipped & simplified by zoom
Open & free for all developers

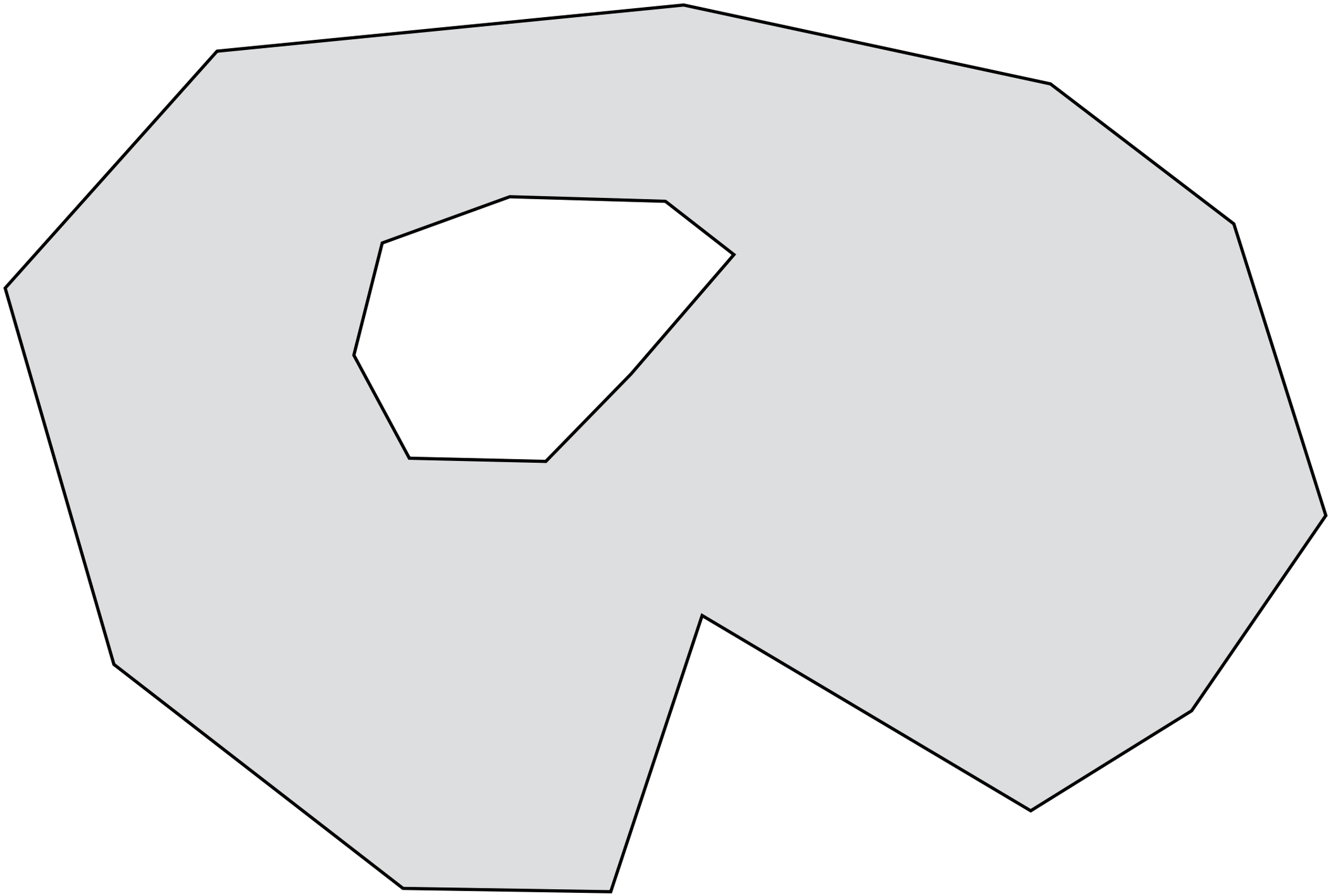
Served in:

- GeoJSON & TopoJSON
- Mapbox binary
- OpenScienceMap binary

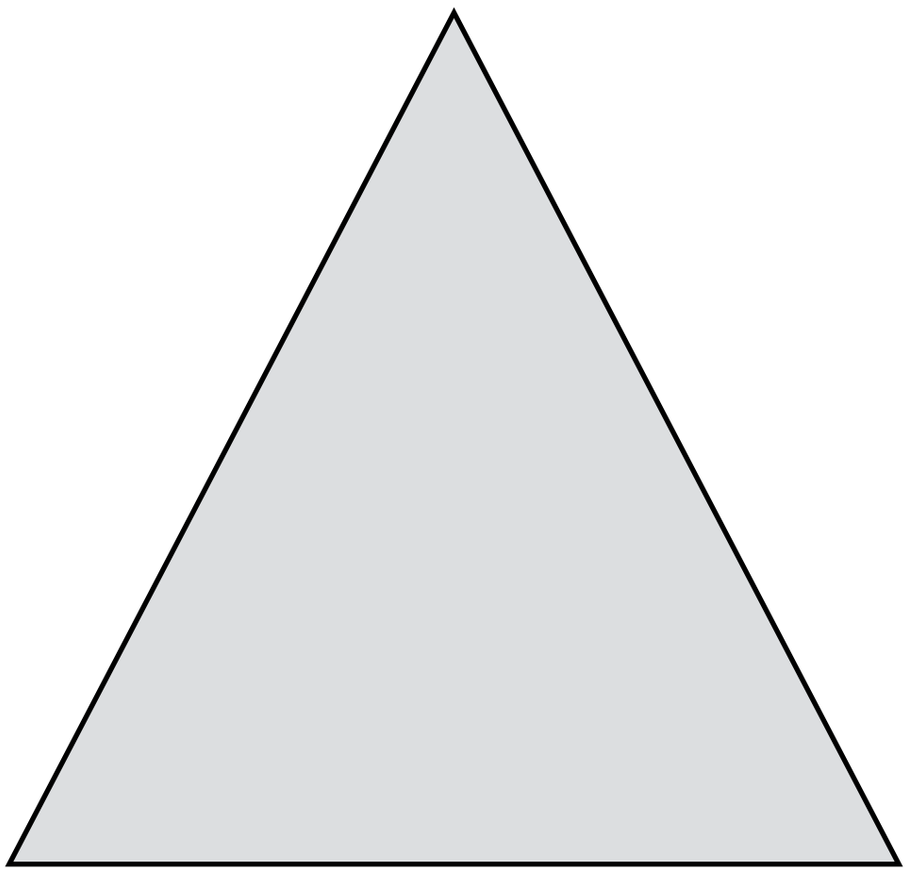
OSM-hosted Vector Tiles

http://openstreetmap.us/~migurski/vector-datasource/

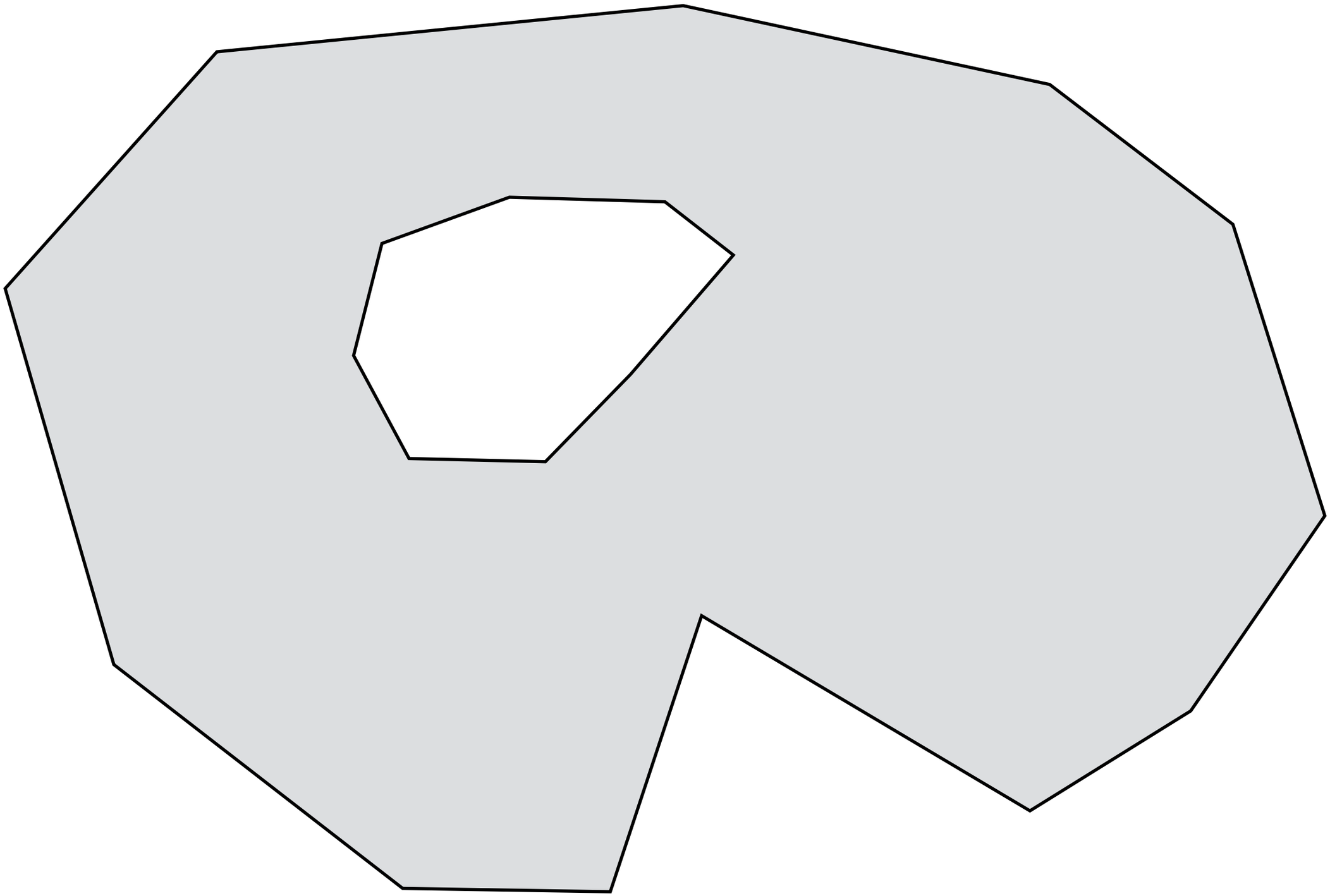
Tessellation: Complex Polygons



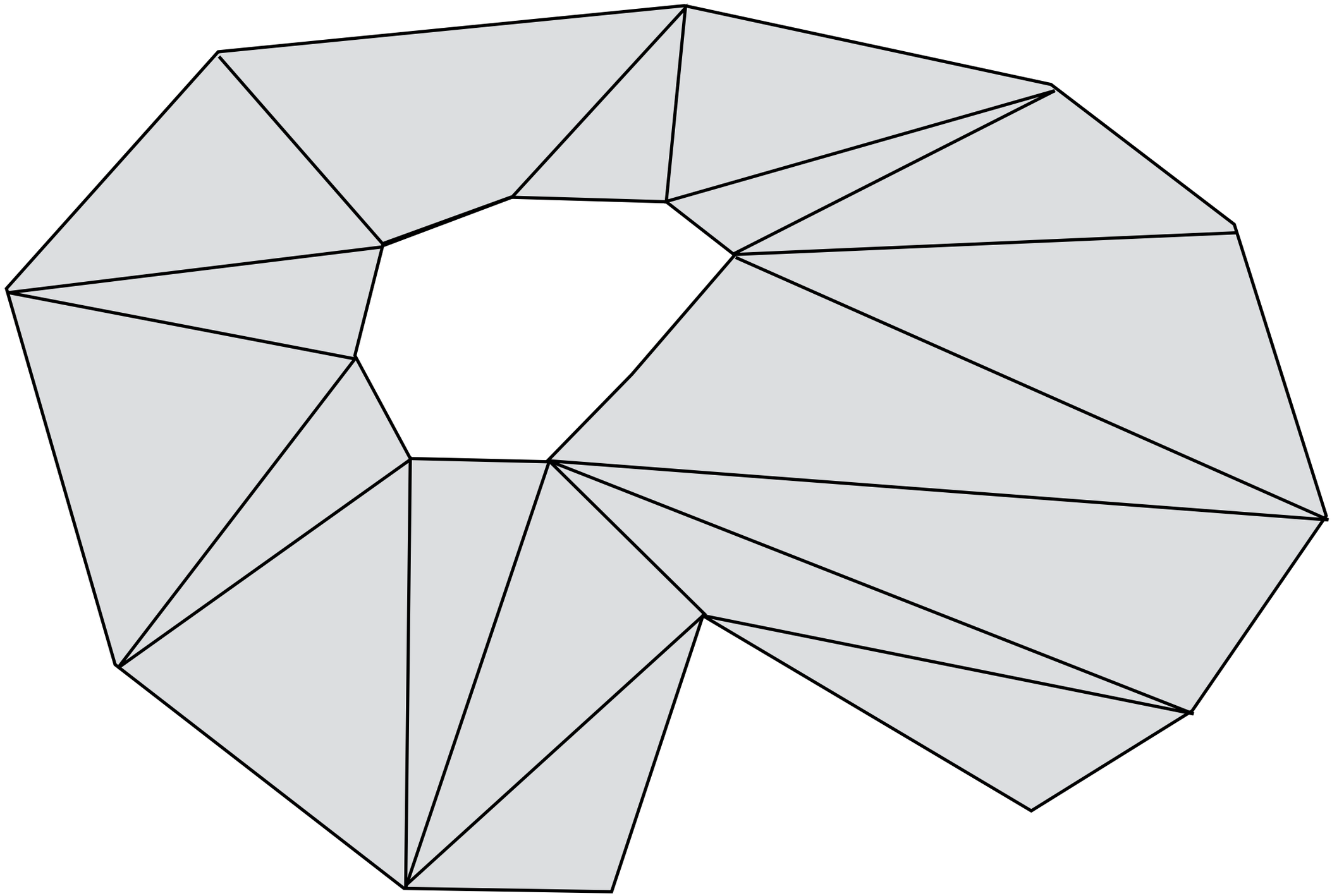
Tessellation: Triangles?!



Tessellation: Complex Polygons => Triangles



Tessellation: Complex Polygons => Triangles



JavaScript Tessellation Libs

https://github.com/brendankenny/libtess.js

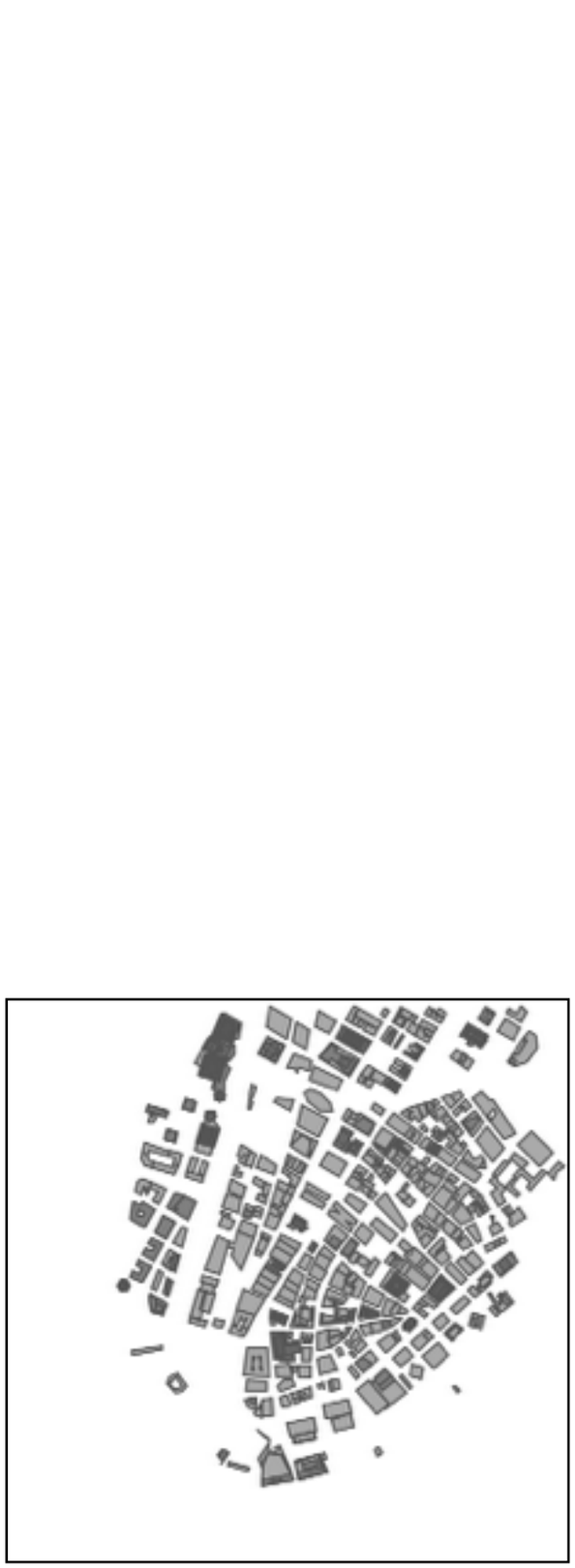
https://github.com/claus/libtess2.js/

https://github.com/cscheid/tessellate

Vertex Shader: Tile Placement

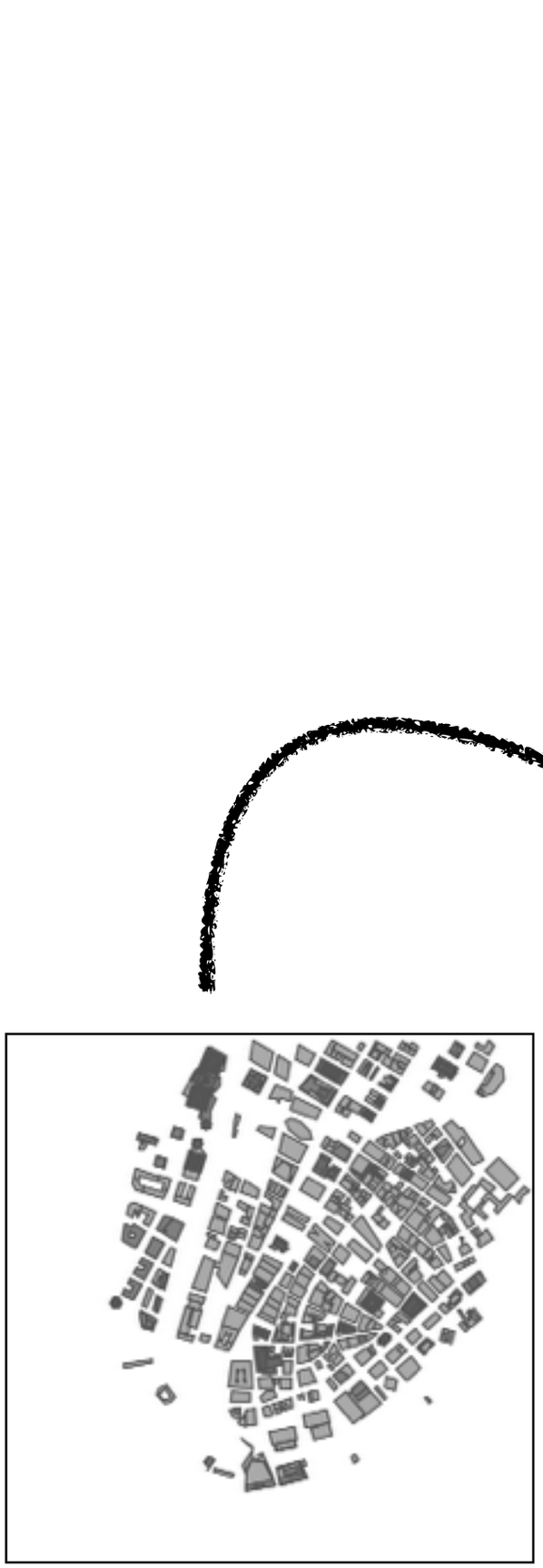
Vertex Shader: Tile Placement

Vertex Shader: Tile Placement



Vertex Shader: Tile Placement

Vertex Shader: Tile Placement



Tangram

a library for visualizing OSM data with WebGL

http://vector-map.mapzen.com

http://github.com/bcamper/tangram

Maps + Shaders

3D Projection

Perspective

Isometric

Maps + Shaders

3D Projection

Perspective

Isometric

Vertex Displacement

Pop-Up

Elevator Buildings

Wave

Maps + Shaders

3D Projection

Perspective

Isometric

Vertex Displacement

Pop-Up

Elevator Buildings

Wave

Procedural Animation

Water

WebGL + OpenGL Resources

http://greggman.github.io/webgl-fundamentals/

http://open.gl/

Thanks!

brett@mapzen.com

http://mapzen.com/