THE UNIVERSITY of NEW MEXICO

Introduction to Computer Graphics with WebGL

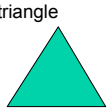
Ed Angel

Fractals and Recursive Subdivision

THE UNIVERSITY of NEW MEXICO

Sierpinski Gasket (2D)

- Start with a triangle



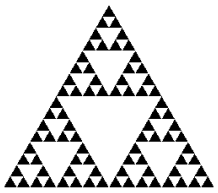
- Connect bisectors of sides and remove central triangle

- Repeat

THE UNIVERSITY of NEW MEXICO

Example

- Five subdivisions



THE UNIVERSITY of NEW MEXICO

The gasket as a fractal

- Consider the filled area (black) and the perimeter (the length of all the lines around the filled triangles)
- As we continue subdividing
  - the area goes to zero
  - but the perimeter goes to infinity
- This is not an ordinary geometric object
  - It is neither two- nor three-dimensional
- It is a *fractal* (fractional dimension) object

THE UNIVERSITY of NEW MEXICO

Gasket Program

- HTML file
  - Same as in other examples
  - Pass through vertex shader
  - Fragment shader sets color
  - Read in JS file

THE UNIVERSITY of NEW MEXICO

Gasket Program

```
var points = [];
var NumTimesToSubdivide = 5;

/* initial triangle */

 var vertices = [
        vec2(-1, -1 ),
        vec2( 0,  1 ),
         vec2( 1, -1 )
    ];

divideTriangle( vertices[0],vertices[1],
     vertices[2], NumTimesToSubdivide);
```

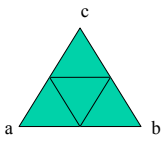
Draw one triangle

```
/* display one triangle  */

function triangle( a, b, c ){
    points.push( a, b, c );
}
```


Triangle Subdivision

```
function divideTriangle( a, b, c, count ){
 // check for end of recursion
    if ( count === 0 ) {
    triangle( a, b, c );
    }
    else {
//bisect the sides
    var ab = mix( a, b, 0.5 );
    var ac = mix( a, c, 0.5 );
    var bc = mix( b, c, 0.5 );
    --count;
// three new triangles
    divideTriangle( a, ab, ac, count-1 );
    divideTriangle( c, ac, bc, count-1 );
    divideTriangle( b, bc, ab, count-1 );
    }
}
```



init()

```
var program = initShaders( gl, "vertex-shader",
    "fragment-shader" );
gl.useProgram( program );
var bufferId = gl.createBuffer();
gl.bindBuffer( gl.ARRAY_BUFFER, bufferId);
gl.bufferData( gl.ARRAY_BUFFER, flatten(points)
     gl.STATIC_DRAW );

var vPosition = gl.getAttribLocation( program,
     "vPosition" );
gl.vertexAttribPointer( vPosition, 2, gl.FLOAT,
     false, 0, 0 );
gl.enableVertexAttribArray( vPosition );

render();
```

THE UNIVERSITY of NEW MEXICO

Render Function

```
function render(){
    gl.clear( gl.COLOR_BUFFER_BIT );
    gl.drawArrays( gl.TRIANGLES, 0, points.length );
}
```