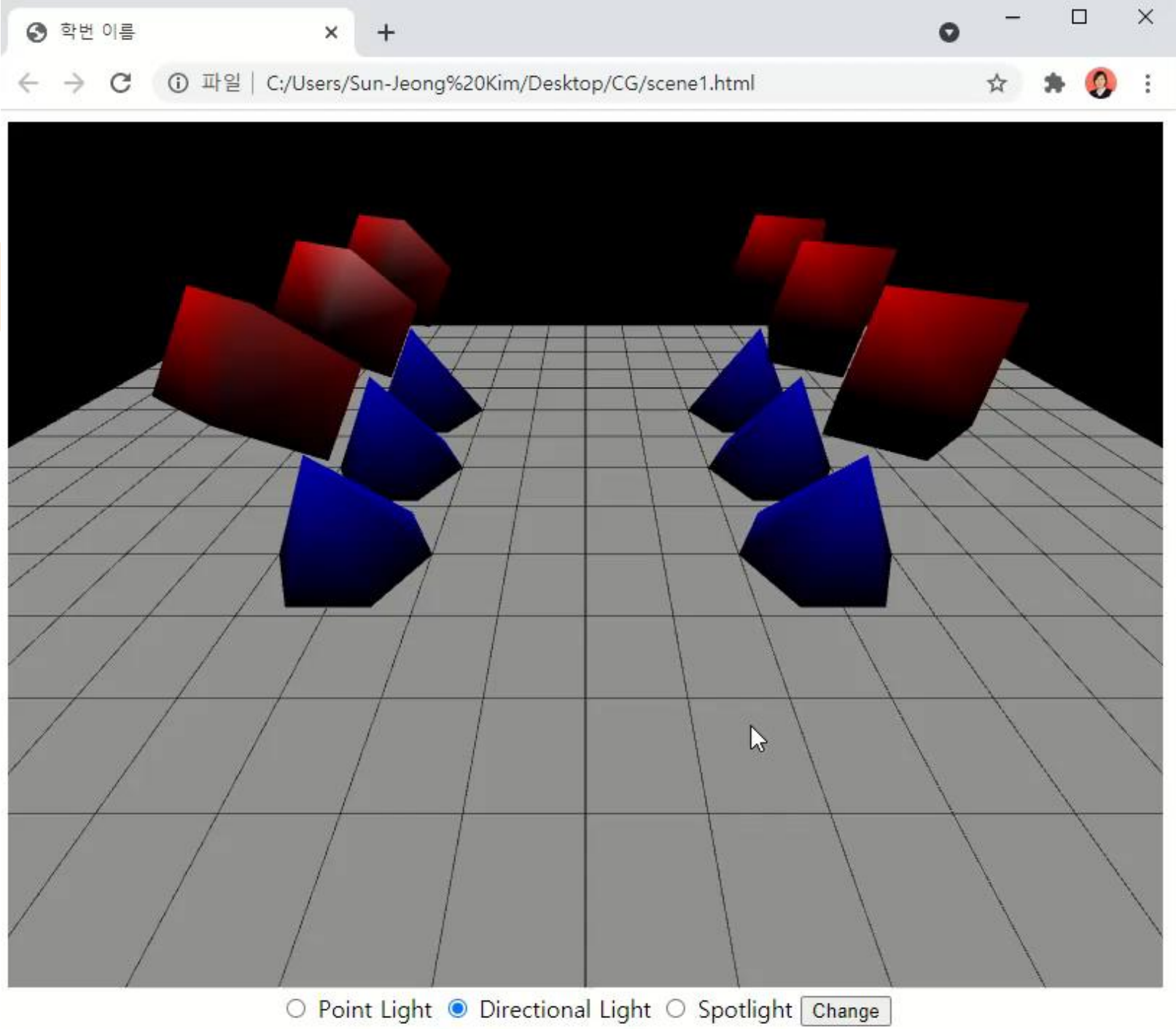
Building a Scene with Light

11TH WEEK, 2021


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene1.html
Point Light
Directional Light
Spotlight
Change

File Edit Selection View Go Run Terminal Help scene1.html - Visual Studio Code

scene1.html × scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.html > html > head > script#vertex-shader

```html
<!DOCTYPE html>
<html>
<head>
    <title>학번 이름</title>
    <script id="vertex-shader" type="x-shader/x-vertex">
        attribute vec4 vPosition;
        attribute vec4 vNormal;
        uniform mat4 modelViewMatrix;
        uniform mat4 projectionMatrix;
        varying vec4 fColor;

        uniform vec4 lightPos, ambientProduct, diffuseProduct, specularProduct;
        uniform vec3 kAtten, spotDir;
        uniform float shininess, spotExp;
        uniform int lighting;   // 0: point, 1: directional, 2: spotlight

        void main() {
            gl_Position = projectionMatrix * modelViewMatrix * vPosition;

            vec3 N = normalize((modelViewMatrix * vNormal).xyz);
            if (lighting == 0) {    // point light
                vec3 pos = (modelViewMatrix * vPosition).xyz;
                vec3 light = lightPos.xyz - pos;
                float d = length(light);
                float atten = 1.0 / (kAtten[0] + kAtten[1]*d + kAtten[2]*d*d);
                vec3 L = normalize(light);
                float kd = max(dot(L, N), 0.0);
                vec4 diffuse = kd * diffuseProduct;

                vec3 V = normalize(-pos);
                vec3 H = normalize(L + V);
                float ks = pow(max(dot(N, H), 0.0), shininess);
                vec4 specular = ks * specularProduct;
```

⊗ 0 ⚠ 0 Ln 15, Col 29 (8 selected) Spaces: 4 UTF-8 CRLF HTML

File Edit Selection View Go Run Terminal Help scene1.html - Visual Studio Code

scene1.html × scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.html > html > head > script#vertex-shader

```
        if (dot(L, N) < 0.0)    specular = vec4(0.0, 0.0, 0.0, 1.0);

        fColor = ambientProduct + atten * (diffuse + specular);
    }
    else if (lighting == 1) {  // directional light
        vec3 L = normalize(lightPos.xyz);
        float kd = max(dot(L, N), 0.0);
        vec4 diffuse = kd * diffuseProduct;

        vec3 V = normalize((modelViewMatrix * vPosition).xyz);  // origin: camera position
        vec3 H = normalize(L - V);
        float ks = pow(max(dot(N, H), 0.0), shininess);
        vec4 specular = ks * specularProduct;

        if (dot(L, N) < 0.0)    specular = vec4(0.0, 0.0, 0.0, 1.0);

        fColor = ambientProduct + diffuse + specular;
    }
    else {      // spotlight
        vec3 pos = (modelViewMatrix * vPosition).xyz;
        vec3 light = lightPos.xyz - pos;
        float d = length(light);
        float atten = 1.0 / (kAtten[0] + kAtten[1]*d + kAtten[2]*d*d);
        vec3 L = normalize(light);
        float kd = max(dot(L, N), 0.0);
        vec4 diffuse = kd * diffuseProduct;

        vec3 V = normalize(-pos);
        vec3 H = normalize(L + V);
        float ks = pow(max(dot(N, H), 0.0), shininess);
        vec4 specular = ks * specularProduct;

```

⊗ 0 ⚠ 0 Ln 15, Col 29 (8 selected) Spaces: 4 UTF-8 CRLF HTML

File Edit Selection View Go Run Terminal Help scene1.html - Visual Studio Code

scene1.html × scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.html > html > head > script#vertex-shader

```
                if (dot(L, N) < 0.0)    specular = vec4(0.0, 0.0, 0.0, 1.0);

                float spotDot = max(dot(normalize(spotDir), -L), 0.0);
                if (spotDot > 0.0)
                    atten *= pow(spotDot, spotExp);
                else
                    atten = 0.0;

                fColor = ambientProduct + atten * (diffuse + specular);

            }
            fColor.a = 1.0;
        }
    </script>

    <script id="fragment-shader" type="x-shader/x-fragment">
        precision mediump float;
        varying vec4 fColor;

        void main() {
            gl_FragColor = fColor;
        }
    </script>

    <script type="text/javascript" src="Common/webgl-utils.js"></script>
    <script type="text/javascript" src="Common/initShaders.js"></script>
    <script type="text/javascript" src="Common/MV.js"></script>
    <script type="text/javascript" src="trackball.js"></script>
    <script type="text/javascript" src="scene1.js"></script>
</head>
    <body>
        <canvas id="gl-canvas" width="800" height="600">
            Oops... your browser doesn't support the HTML5 canvas element!
```

⊗ 0 ⚠ 0 Ln 15, Col 29 (8 selected) Spaces: 4 UTF-8 CRLF HTML

File Edit Selection View Go Run Terminal Help scene1.html - Visual Studio Code

scene1.html scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.html > html > head > script#vertex-shader

```html

    <script type="text/javascript" src="Common/webgl-utils.js"></script>
    <script type="text/javascript" src="Common/initShaders.js"></script>
    <script type="text/javascript" src="Common/MV.js"></script>
    <script type="text/javascript" src="trackball.js"></script>
    <script type="text/javascript" src="scene1.js"></script>
</head>
    <body>
        <canvas id="gl-canvas" width="800" height="600">
            Oops... your browser doesn't support the HTML5 canvas element!
        </canvas><br>
        <div style="width: 800px; text-align: center;">
            <input type="radio" id="point" name="lighting"> Point Light
            <input type="radio" id="directional" name="lighting" checked> Directional Light
            <input type="radio" id="spotlight" name="lighting"> Spotlight
            <button id="change">Change</button>
        </div>
    </body>
</html>
```

Ln 15, Col 29 (8 selected) Spaces: 4 UTF-8 CRLF HTML

File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

scene1.html | scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > render

```javascript
var gl;
var points = [];
var normals = [];

var theta = 0.0;

var viewMatrix;
var modelViewMatrixLoc, lightingLoc, diffuseProductLoc;

var trballMatrix = mat4(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);

var numVertCubeTri, numVertPyraTri, numVertGroundTri, numVertGroundLine;

var lighting = 1;   // 0: point, 1: directional, 2: spotlight

window.onload = function init()
{
    var canvas = document.getElementById("gl-canvas");

    gl = WebGLUtils.setupWebGL(canvas);
    if( !gl ) {
        alert("WebGL isn't available!");
    }

    generateCube();
    generateHexaPyramid();
    generateGround(10);

    // virtual trackball
    var trball = trackball(canvas.width, canvas.height);
    var mouseDown = false;

    canvas.addEventListener("mousedown", function (event) {
```

⊗ 0 ⚠ 0 Ln 194, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

scene1.html | scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > render

```javascript
    canvas.addEventListener("mousedown", function (event) {
        trball.start(event.clientX, event.clientY);

        mouseDown = true;
    });

    canvas.addEventListener("mouseup", function (event) {
        mouseDown = false;
    });

    canvas.addEventListener("mousemove", function (event) {
        if (mouseDown) {
            trball.end(event.clientX, event.clientY);

            trballMatrix = mat4(trball.rotationMatrix);
        }
    });

    // Configure WebGL
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(0.0, 0.0, 0.0, 1.0);

    // Enable hidden-surface removal
    gl.enable(gl.DEPTH_TEST);

    gl.enable(gl.POLYGON_OFFSET_FILL);
    gl.polygonOffset(0.01, 1);

    // Load shaders and initialize attribute buffers
    var program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);

    // Load the data into the GPU
```

⊗ 0 ⚠ 0 Ln 194, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

scene1.html | scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > render

```javascript
    // Load the data into the GPU
    var bufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);

    // Associate our shader variables with our data buffer
    var vPosition = gl.getAttribLocation(program, "vPosition");
    gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vPosition);

    // Create a buffer object, initialize it, and associate it with
    // the associated attribute variable in our vertex shader
    var nBufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, nBufferId);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(normals), gl.STATIC_DRAW);

    var vNormal = gl.getAttribLocation(program, "vNormal");
    gl.vertexAttribPointer(vNormal, 4, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vNormal);

    viewMatrix = lookAt(vec3(0, 3, 3), vec3(0, 0, 0), vec3(0, 1, 0));
    modelViewMatrixLoc = gl.getUniformLocation(program, "modelViewMatrix");
    /*
    // 3D orthographic viewing
    var viewLength = 2.0;
    if (canvas.width > canvas.height) {
        var aspect = viewLength * canvas.width / canvas.height;
        projectionMatrix = ortho(-aspect, aspect, -viewLength, viewLength, -viewLength, 1000);
    }
    else {
        var aspect = viewLength * canvas.height / canvas.width;
        projectionMatrix = ortho(-viewLength, viewLength, -aspect, aspect, -viewLength, 1000);
    }
```

⊗ 0 ⚠ 0 Ln 194, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

scene1.html | scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > render

```javascript
    */
    // 3D perspective viewing
    var aspect = canvas.width / canvas.height;
    projectionMatrix = perspective(90, aspect, 0.1, 1000);
    projectionMatrixLoc = gl.getUniformLocation(program, "projectionMatrix");
    gl.uniformMatrix4fv(projectionMatrixLoc, false, flatten(projectionMatrix));

    lightingLoc = gl.getUniformLocation(program, "lighting");
    gl.uniform1i(lightingLoc, lighting);

    // Event listeners for buttons
    document.getElementById("change").onclick = function () {
        if (document.getElementById("point").checked)
            lighting = 0;
        else if(document.getElementById("directional").checked)
            lighting = 1;
        else
            lighting = 2;

        gl.uniform1i(lightingLoc, lighting);
    };

    setLighting(program);

    render();
};

function setLighting(program) {
    var lightPos = [0.0, 1.0, 0.0, 0.0];
    var lightAmbient = [0.2, 0.2, 0.2, 1.0];
    var lightDiffuse = [1.0, 1.0, 1.0, 1.0];
    var lightSpecular = [1.0, 1.0, 1.0, 1.0];

```

⊗ 0 ⚠ 0 Ln 194, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

scene1.html | scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > render

```javascript
    var matAmbient = [1.0, 1.0, 1.0, 1.0];
    var matDiffuse = [1.0, 1.0, 1.0, 1.0];
    var matSpecular = [1.0, 1.0, 1.0, 1.0];

    var ambientProduct = mult(lightAmbient, matAmbient);
    var diffuseProduct = mult(lightDiffuse, matDiffuse);
    var specularProduct = mult(lightSpecular, matSpecular);

    gl.uniform4fv(gl.getUniformLocation(program, "lightPos"), lightPos);
    gl.uniform4fv(gl.getUniformLocation(program, "ambientProduct"), ambientProduct);
    diffuseProductLoc = gl.getUniformLocation(program, "diffuseProduct");
    gl.uniform4fv(diffuseProductLoc, diffuseProduct);
    gl.uniform4fv(gl.getUniformLocation(program, "specularProduct"), specularProduct);

    gl.uniform3f(gl.getUniformLocation(program, "kAtten"), 1.0, 0.1, 0.1);
    gl.uniform3f(gl.getUniformLocation(program, "spotDir"), 0.0, -1.0, -1.0);
    gl.uniform1f(gl.getUniformLocation(program, "spotExp"), 5.0);
    gl.uniform1f(gl.getUniformLocation(program, "shininess"), 100.0);
}

function render() {
    gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);

    theta += 2.0;

    for (var z=-5; z<0; z+=2) {
        // draw a cube
        gl.uniform4f(diffuseProductLoc, 1.0, 0.0, 0.0, 1.0);    // red

        var rMatrix = mult(rotateY(theta), rotateZ(45));
        var modelMatrix = mult(translate(-3, 1.3, z), rMatrix);
        modelMatrix = mult(trballMatrix, modelMatrix);
        modelViewMatrix = mult(viewMatrix, modelMatrix);
```

⊗ 0 ⚠ 0 Ln 194, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

scene1.html | scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > render

```javascript
        modelViewMatrix = mult(viewMatrix, modelMatrix);
        gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
        gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

        modelMatrix = mult(translate(3, 1.3, z), rMatrix);
        modelMatrix = mult(trballMatrix, modelMatrix);
        modelViewMatrix = mult(viewMatrix, modelMatrix);
        gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
        gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

        // draw a hexa-pyramid
        gl.uniform4f(diffuseProductLoc, 0.0, 0.0, 1.0, 1.0);    // blue

        modelMatrix = mult(translate(-3, -0.5, z), rotateZ(180));
        modelMatrix = mult(trballMatrix, modelMatrix);
        modelViewMatrix = mult(viewMatrix, modelMatrix);
        gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
        gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);

        modelMatrix = mult(translate(3, -0.5, z), rotateZ(180));
        modelMatrix = mult(trballMatrix, modelMatrix);
        modelViewMatrix = mult(viewMatrix, modelMatrix);
        gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
        gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);
    }

    // draw the ground
    gl.uniform4f(diffuseProductLoc, 0.8, 0.8, 0.8, 1.0);    // gray

    modelViewMatrix = mult(viewMatrix, trballMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    //gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri+numVertPyraTri, numVertGroundTri);
```

⊗ 0 ⚠ 0 Ln 194, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

scene1.html | scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > render

```js
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri+numVertPyraTri, numVertGroundTri);
    gl.drawArrays(gl.LINES, numVertCubeTri+numVertPyraTri+numVertGroundTri, numVertGroundLine);

    requestAnimationFrame(render);
}

function generateCube() {
    numVertCubeTri = 0;
    quad(1, 0, 3, 2);
    quad(2, 3, 7, 6);
    quad(3, 0, 4, 7);
    quad(4, 5, 6, 7);
    quad(5, 4, 0, 1);
    quad(6, 5, 1, 2);
}

function quad(a, b, c, d) {
    const vertexPos = [
        vec4(-0.5, -0.5, -0.5, 1.0),
        vec4( 0.5, -0.5, -0.5, 1.0),
        vec4( 0.5,  0.5, -0.5, 1.0),
        vec4(-0.5,  0.5, -0.5, 1.0),
        vec4(-0.5, -0.5,  0.5, 1.0),
        vec4( 0.5, -0.5,  0.5, 1.0),
        vec4( 0.5,  0.5,  0.5, 1.0),
        vec4(-0.5,  0.5,  0.5, 1.0)
    ];

    const vertexNormal = [
        vec4(-0.57735, -0.57735, -0.57735, 0.0),
        vec4( 0.57735, -0.57735, -0.57735, 0.0),
        vec4( 0.57735,  0.57735, -0.57735, 0.0),
        vec4(-0.57735,  0.57735, -0.57735, 0.0),
```

⊗ 0 ⚠ 0 Ln 194, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

scene1.html | scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > render

```
        vec4(-0.57735, -0.57735,  0.57735, 0.0),
        vec4( 0.57735, -0.57735,  0.57735, 0.0),
        vec4( 0.57735,  0.57735,  0.57735, 0.0),
        vec4(-0.57735,  0.57735,  0.57735, 0.0)
    ];

    points.push(vertexPos[a]);
    normals.push(vertexNormal[a]);
    numVertCubeTri++;
    points.push(vertexPos[b]);
    normals.push(vertexNormal[b]);
    numVertCubeTri++;
    points.push(vertexPos[c]);
    normals.push(vertexNormal[c]);
    numVertCubeTri++;

    points.push(vertexPos[a]);
    normals.push(vertexNormal[a]);
    numVertCubeTri++;
    points.push(vertexPos[c]);
    normals.push(vertexNormal[c]);
    numVertCubeTri++;
    points.push(vertexPos[d]);
    normals.push(vertexNormal[d]);
    numVertCubeTri++;
}

function generateGround(scale) {
    numVertGroundTri = 0;
    // two triangles
    points.push(vec4(scale, -1.0, -scale, 1.0));
    normals.push(vec4(0.0, 1.0, 0.0, 0.0));
    numVertGroundTri++;
```

⊗ 0 ⚠ 0 Ln 194, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

scene1.html | scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > render

```js
    numVertGroundTri++;
    points.push(vec4(-scale, -1.0, -scale, 1.0));
    normals.push(vec4(0.0, 1.0, 0.0, 0.0));
    numVertGroundTri++;
    points.push(vec4(-scale, -1.0, scale, 1.0));
    normals.push(vec4(0.0, 1.0, 0.0, 0.0));
    numVertGroundTri++;

    points.push(vec4(scale, -1.0, -scale, 1.0));
    normals.push(vec4(0.0, 1.0, 0.0, 0.0));
    numVertGroundTri++;
    points.push(vec4(-scale, -1.0, scale, 1.0));
    normals.push(vec4(0.0, 1.0, 0.0, 0.0));
    numVertGroundTri++;
    points.push(vec4(scale, -1.0, scale, 1.0));
    normals.push(vec4(0.0, 1.0, 0.0, 0.0));
    numVertGroundTri++;

    numVertGroundLine = 0;
    // grid lines
    for(var x=-scale; x<=scale; x++) {
        points.push(vec4(x, -1.0, -scale, 1.0));
        normals.push(vec4(0.0, 0.0, 0.0, 0.0));
        numVertGroundLine++;
        points.push(vec4(x, -1.0, scale, 1.0));
        normals.push(vec4(0.0, 0.0, 0.0, 0.0));
        numVertGroundLine++;
    }
    for(var z=-scale; z<=scale; z++) {
        points.push(vec4(-scale, -1.0, z, 1.0));
        normals.push(vec4(0.0, 0.0, 0.0, 0.0));
        numVertGroundLine++;
        points.push(vec4(scale, -1.0, z, 1.0));
```

⊗ 0 ⚠ 0 Ln 194, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

scene1.html | scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > render

```
        points.push(vec4(scale, -1.0, z, 1.0));
        normals.push(vec4(0.0, 0.0, 0.0, 0.0));
        numVertGroundLine++;
    }
}

function generateHexaPyramid() {
    const vertexPos = [
        vec4(0.0, 0.5, 0.0, 1.0),
        vec4(1.0, 0.5, 0.0, 1.0),
        vec4(0.5, 0.5, -0.866, 1.0),
        vec4(-0.5, 0.5, -0.866, 1.0),
        vec4(-1.0, 0.5, 0.0, 1.0),
        vec4(-0.5, 0.5, 0.866, 1.0),
        vec4(0.5, 0.5, 0.866, 1.0),
        vec4(0.0, -1.0, 0.0, 1.0)
    ];

    const vertexNormal = [
        vec4(0.0, 1.0, 0.0, 0.0),
        vec4(1.0, 0.0, 0.0, 0.0),
        vec4(0.5, 0.0, -0.866, 0.0),
        vec4(-0.5, 0.0, -0.866, 0.0),
        vec4(-1.0, 0.0, 0.0, 0.0),
        vec4(-0.5, 0.0, 0.866, 0.0),
        vec4(0.5, 0.0, 0.866, 0.0),
        vec4(0.0, -1.0, 0.0, 0.0)
    ];

    numVertPyraTri = 0;
    for (var i=1; i<6; i++) {
        points.push(vertexPos[0]);
        normals.push(vertexNormal[0]);
```

⊗ 0 ⚠ 0 Ln 194, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

scene1.html | scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > render

```
            normals.push(vertexNormal[0]);
            numVertPyraTri++;
            points.push(vertexPos[i]);
            normals.push(vertexNormal[0]);
            numVertPyraTri++;
            points.push(vertexPos[i+1]);
            normals.push(vertexNormal[0]);
            numVertPyraTri++;

            points.push(vertexPos[7]);
            normals.push(vertexNormal[7]);
            numVertPyraTri++;
            points.push(vertexPos[i+1]);
            normals.push(vertexNormal[i+1]);
            numVertPyraTri++;
            points.push(vertexPos[i]);
            normals.push(vertexNormal[i]);
            numVertPyraTri++;
        }
        points.push(vertexPos[0]);
        normals.push(vertexNormal[0]);
        numVertPyraTri++;
        points.push(vertexPos[6]);
        normals.push(vertexNormal[0]);
        numVertPyraTri++;
        points.push(vertexPos[1]);
        normals.push(vertexNormal[0]);
        numVertPyraTri++;

        points.push(vertexPos[7]);
        normals.push(vertexNormal[7]);
        numVertPyraTri++;
        points.push(vertexPos[1]);
```

⊗ 0 ⚠ 0 Ln 194, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

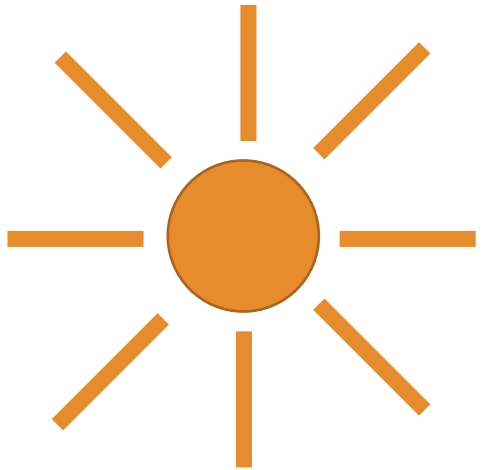
scene1.html | scene1.js

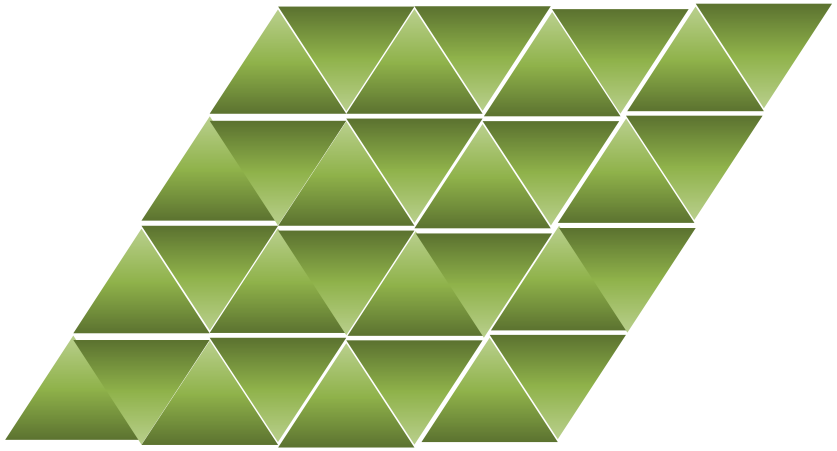
C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > render

```
        points.push(vertexPos[0]);
        normals.push(vertexNormal[0]);
        numVertPyraTri++;
        points.push(vertexPos[6]);
        normals.push(vertexNormal[0]);
        numVertPyraTri++;
        points.push(vertexPos[1]);
        normals.push(vertexNormal[0]);
        numVertPyraTri++;

        points.push(vertexPos[7]);
        normals.push(vertexNormal[7]);
        numVertPyraTri++;
        points.push(vertexPos[1]);
        normals.push(vertexNormal[1]);
        numVertPyraTri++;
        points.push(vertexPos[6]);
        normals.push(vertexNormal[6]);
        numVertPyraTri++;
    }

```

⊗ 0 ⚠ 0 Ln 194, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

What’s the Difference?





Ground

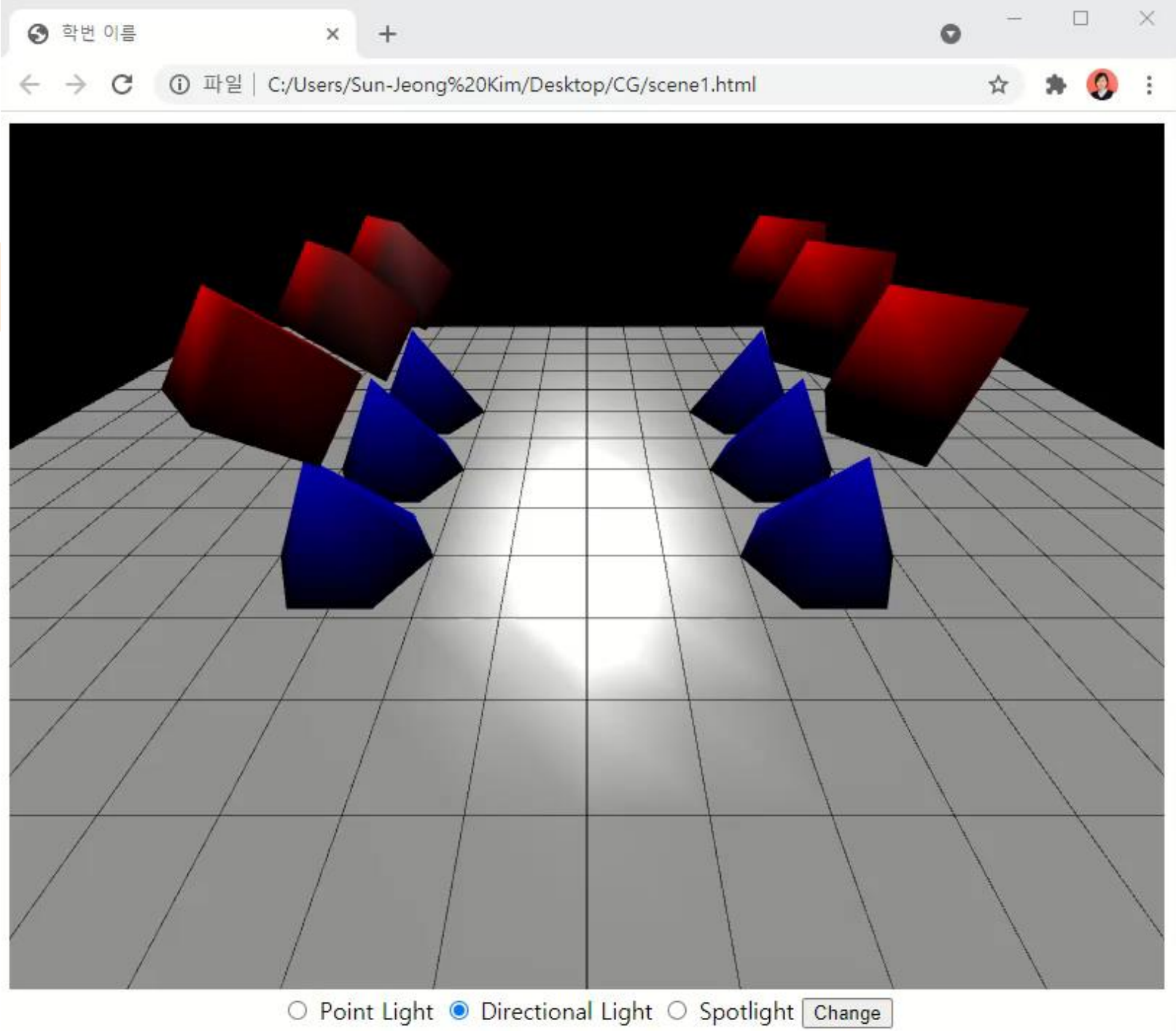
File Edit Selection View Go Run Terminal Help scene1.js - Visual Studio Code

scene1.html | scene1.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene1.js > generateGround

```
function generateGround(scale) {
    numVertGroundTri = 0;
    for(var x=-scale; x<scale; x++) {
        for(var z=-scale; z<scale; z++) {
            // two triangles
            points.push(vec4(x, -1.0, z, 1.0));
            normals.push(vec4(0.0, 1.0, 0.0, 0.0));
            numVertGroundTri++;
            points.push(vec4(x, -1.0, z+1, 1.0));
            normals.push(vec4(0.0, 1.0, 0.0, 0.0));
            numVertGroundTri++;
            points.push(vec4(x+1, -1.0, z+1, 1.0));
            normals.push(vec4(0.0, 1.0, 0.0, 0.0));
            numVertGroundTri++;

            points.push(vec4(x, -1.0, z, 1.0));
            normals.push(vec4(0.0, 1.0, 0.0, 0.0));
            numVertGroundTri++;
            points.push(vec4(x+1, -1.0, z+1, 1.0));
            normals.push(vec4(0.0, 1.0, 0.0, 0.0));
            numVertGroundTri++;
            points.push(vec4(x+1, -1.0, z, 1.0));
            normals.push(vec4(0.0, 1.0, 0.0, 0.0));
            numVertGroundTri++;
        }
    }

    numVertGroundLine = 0;
    // grid lines
    for(var x=-scale; x<=scale; x++) {
        points.push(vec4(x, -1.0, -scale, 1.0));
        normals.push(vec4(0.0, 0.0, 0.0, 0.0));
```

⊗ 0 ⚠ 0


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene1.html
Point Light
Directional Light
Spotlight
Change

연습 문제 (1)

- Directional 조명을 선택할 때,
  scene1.js에 있는 setLighting( ) 함수에서,
  - lightPos의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) var lightPos = vec4(1.0, 0.0, 0.0, 0.0);

  - lightAmbient, lightDiffuse, lightSpecular의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) var lightAmbient = vec4(1.0, 0.0, 0.0, 1.0);
      var lightDiffuse = vec4(0.0, 1.0, 0.0, 1.0);
      var lightSpecular = vec4(0.0, 0.0, 1.0, 1.0);

연습 문제 (2)

- Point 조명을 선택 후
  light.js에 있는 setLighting( ) 함수에서,
  - lightPos의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) var lightPos = vec4(1.0, 0.0, 0.0, 1.0);

  - kAtten의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) gl.uniform3f(attenLoc, 1.0, 0.0, 0.0);

연습 문제 (3)

- Spotlight 선택 후
  light.js에 있는 setLighting( ) 함수에서,
  - spotDir의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) gl.uniform3f(spotDirLoc, 0.0, 0.0, -1.0);

  - spotExp의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) gl.uniform1f(spotExpLoc, 10.0);