

# 렌더링 비교

12<sup>TH</sup> WEEK, 2021





&lt;&gt; scene\_smooth\_dir.html × JS scene\_smooth\_dir.js



C: &gt; Users &gt; Sun-Jeong Kim &gt; Desktop &gt; CG &gt; &lt;&gt; scene\_smooth\_dir.html &gt; html &gt; head &gt; script

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Gouraud + Directional</title>
5      <script id="vertex-shader" type="x-shader/x-vertex">
6          attribute vec4 vPosition;
7          attribute vec4 vNormal;
8          uniform mat4 modelViewMatrix;
9          uniform mat4 projectionMatrix;
10         varying vec4 fColor;
11
12         uniform vec4 lightPos, ambientProduct, diffuseProduct, specularProduct;
13         uniform float shininess;
14
15         void main() {
16             gl_Position = projectionMatrix * modelViewMatrix * vPosition;
17
18             vec3 N = normalize((modelViewMatrix * vNormal).xyz);
19             vec3 L = normalize(lightPos.xyz);
20             float kd = max(dot(L, N), 0.0);
21             vec4 diffuse = kd * diffuseProduct;
22
23             vec3 V = normalize((modelViewMatrix * vPosition).xyz); // origin: camera position
24             vec3 H = normalize(L - V);
25             float ks = pow(max(dot(N, H), 0.0), shininess);
26             vec4 specular = ks * specularProduct;
27
28             if (dot(L, N) < 0.0)    specular = vec4(0.0, 0.0, 0.0, 1.0);
29
30             fColor = ambientProduct + diffuse + specular;
31             fColor.a = 1.0;
32         }
33     </script>
```

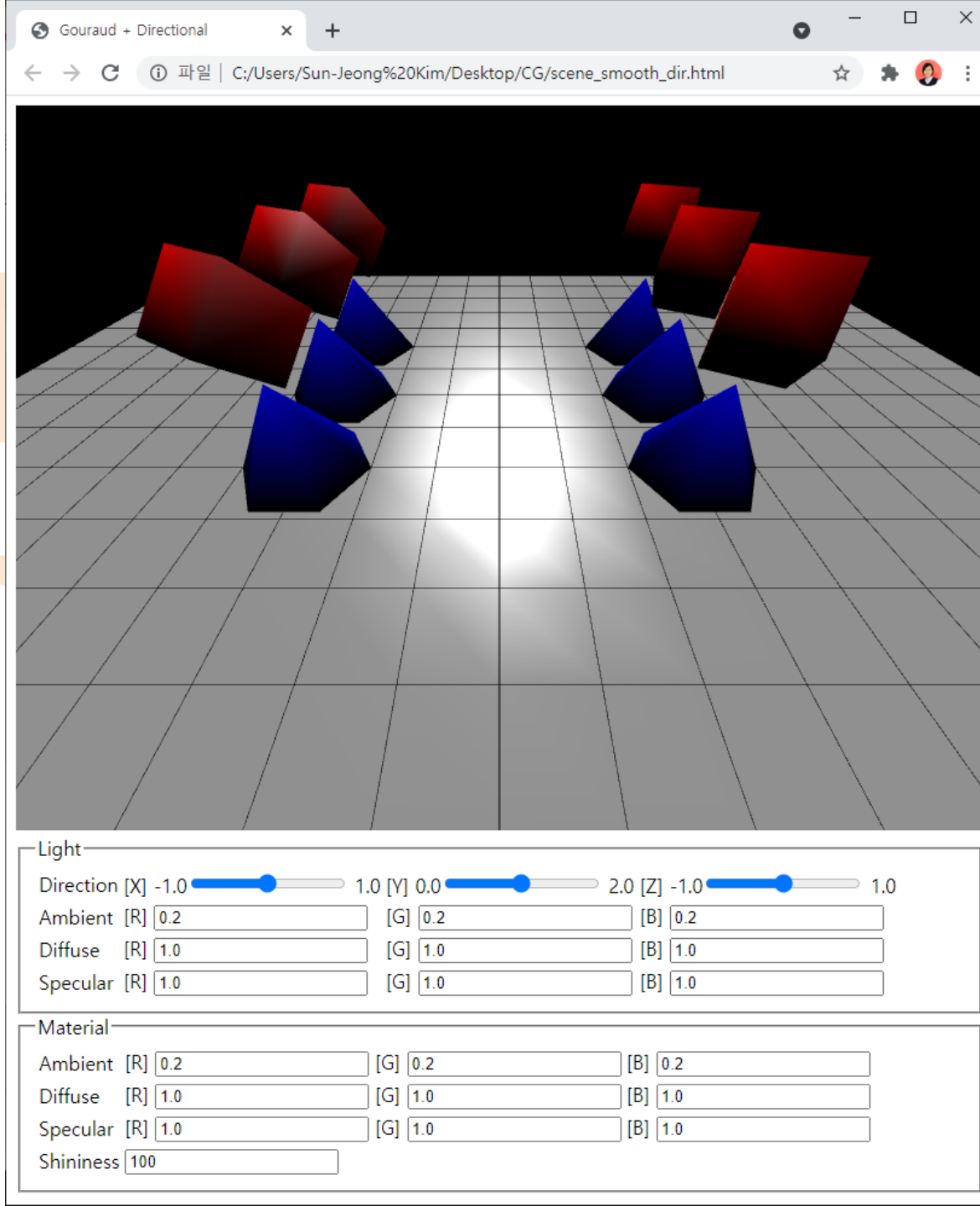


&lt;&gt; scene\_smooth\_dir.html JS scene\_smooth\_dir.js X

C: &gt; Users &gt; Sun-Jeong Kim &gt; Desktop &gt; CG &gt; JS scene\_smooth\_dir.js &gt; ...

```
1  var gl;
2  var points = [];
3  var normals = [];
4
5  var theta = 0.0;
6
7  var viewMatrix;
8  var modelViewMatrixLoc;
9  var lightPosLoc, ambientProductLoc, diffuseProductLoc, specularProductLoc, shininessLoc;
10
11 var trballMatrix = mat4(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);
12
13 var numVertCubeTri, numVertPyraTri, numVertGroundTri, numVertGroundLine;
14
15 window.onload = function init()
16 {
17     var canvas = document.getElementById("gl-canvas");
18
19     gl = WebGLUtils.setupWebGL(canvas);
20     if( !gl ) {
21         alert("WebGL isn't available!");
22     }
23
24     generateCube();
25     generateHexaPyramid();
26     generateGround(10);
27
28     // virtual trackball
29     var trball = trackball(canvas.width, canvas.height);
30     var mouseDown = false;
31
32     canvas.addEventListener("mousedown", function (event) {
33         trball.start(event.clientX, event.clientY);
```





&lt;&gt; scene\_smooth\_dir.html JS scene\_smooth\_dir.js &lt;&gt; scene\_smooth\_point.html JS scene\_smooth\_point.js X

C: &gt; Users &gt; Sun-Jeong Kim &gt; Desktop &gt; CG &gt; JS scene\_smooth\_point.js &gt; setLighting

```
1  var gl;
2  var points = [];
3  var normals = [];
4
5  var theta = 0.0;
6
7  var viewMatrix;
8  var modelViewMatrixLoc;
9  var lightPosLoc, ambientProductLoc, diffuseProductLoc, specularProductLoc, shininessLoc;
10 var lightAttenLoc;
11
12 var trballMatrix = mat4(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);
13
14 var numVertCubeTri, numVertPyraTri, numVertGroundTri, numVertGroundLine;
15
16 window.onload = function init()
17 {
18     var canvas = document.getElementById("gl-canvas");
19
20     gl = WebGLUtils.setupWebGL(canvas);
21     if( !gl ) {
22         alert("WebGL isn't available!");
23     }
24
25     generateCube();
26     generateHexaPyramid();
27     generateGround(10);
28
29     // virtual trackball
30     var trball = trackball(canvas.width, canvas.height);
31     var mouseDown = false;
32
33     canvas.addEventListener("mousedown", function (event) {
```

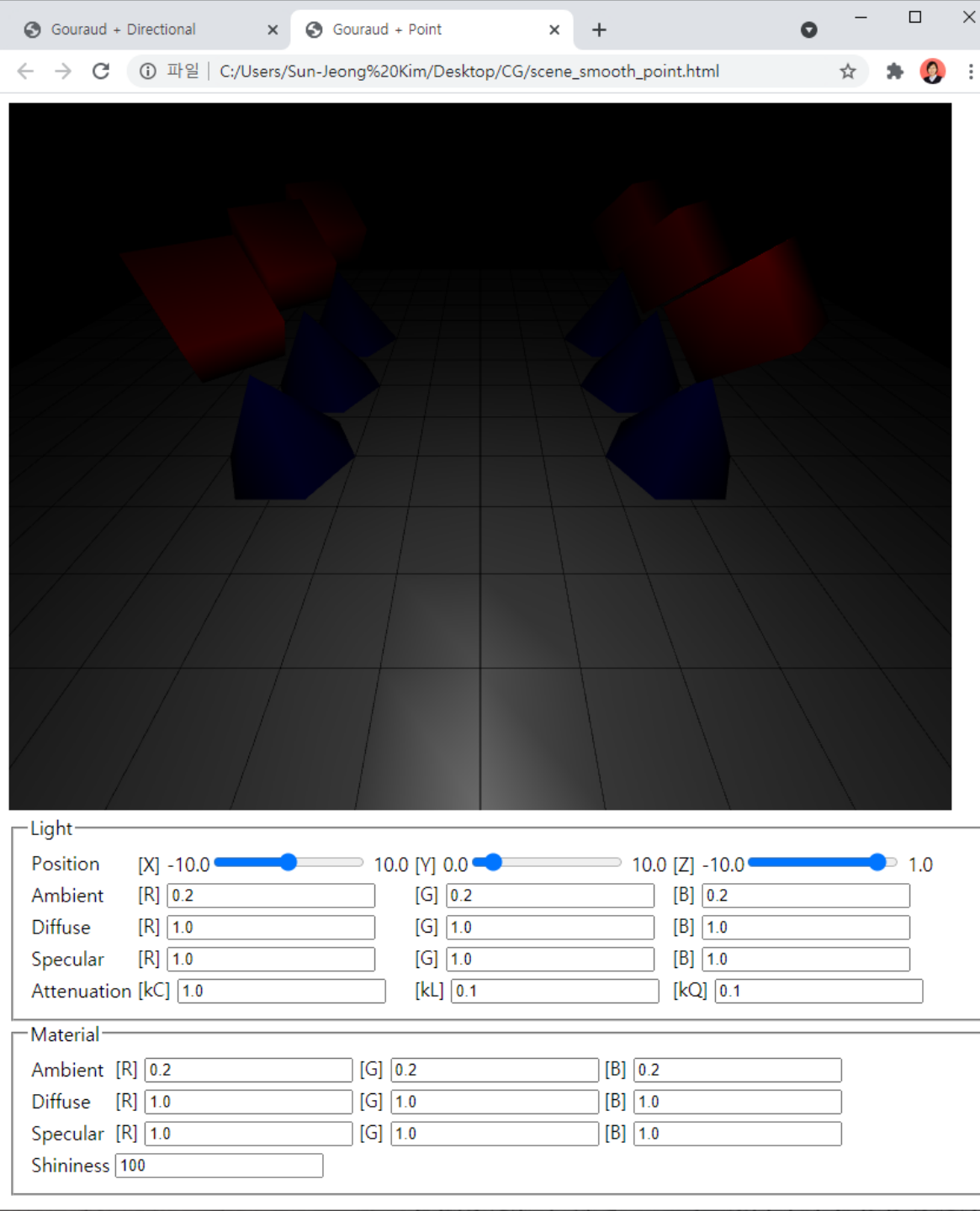


&lt;&gt; scene\_smooth\_dir.html JS scene\_smooth\_dir.js &lt;&gt; scene\_smooth\_point.html JS scene\_smooth\_point.js X

C: &gt; Users &gt; Sun-Jeong Kim &gt; Desktop &gt; CG &gt; JS scene\_smooth\_point.js &gt; init

```
1  var gl;
2  var points = [];
3  var normals = [];
4
5  var theta = 0.0;
6
7  var viewMatrix;
8  var modelViewMatrixLoc;
9  var lightPosLoc, ambientProductLoc, diffuseProductLoc, specularProductLoc, shininessLoc;
10 var lightAttenLoc;
11
12 var trballMatrix = mat4(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);
13
14 var numVertCubeTri, numVertPyraTri, numVertGroundTri, numVertGroundLine;
15
16 window.onload = function init()
17 {
18     var canvas = document.getElementById("gl-canvas");
19
20     gl = WebGLUtils.setupWebGL(canvas);
21     if( !gl ) {
22         alert("WebGL isn't available!");
23     }
24
25     generateCube();
26     generateHexaPyramid();
27     generateGround(10);
28
29     // virtual trackball
30     var trball = trackball(canvas.width, canvas.height);
31     var mouseDown = false;
32
33     canvas.addEventListener("mousedown", function (event) {
```









scene\_smooth\_dir.html

JS scene\_smooth\_dir.js

&lt;&gt; scene\_smooth\_point.html

JS scene\_smooth\_point.js

&lt;&gt; scene\_smooth\_spot.html ×

JS scene\_smooth\_spot.js



C: &gt; Users &gt; Sun-Jeong Kim &gt; Desktop &gt; CG &gt; &lt;&gt; scene\_smooth\_spot.html &gt; html &gt; body &gt; fieldset &gt; table &gt; tr &gt; td

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Gouraud + Spotlight</title>
5 <script id="vertex-shader" type="x-shader/x-vertex">
6     attribute vec4 vPosition;
7     attribute vec4 vNormal;
8     uniform mat4 modelViewMatrix;
9     uniform mat4 projectionMatrix;
10    varying vec4 fColor;
11
12    uniform vec4 lightPos, ambientProduct, diffuseProduct, specularProduct;
13    uniform vec3 kAtten, spotDir;
14    uniform float shininess, spotExp;
15
16    void main() {
17        gl_Position = projectionMatrix * modelViewMatrix * vPosition;
18
19        vec3 N = normalize((modelViewMatrix * vNormal).xyz);
20        vec3 pos = (modelViewMatrix * vPosition).xyz;
21        vec3 light = lightPos.xyz - pos;
22        float d = length(light);
23        float atten = 1.0 / (kAtten[0] + kAtten[1]*d + kAtten[2]*d*d);
24        vec3 L = normalize(light);
25        float kd = max(dot(L, N), 0.0);
26        vec4 diffuse = kd * diffuseProduct;
27
28        vec3 V = normalize(-pos);
29        vec3 H = normalize(L + V);
30        float ks = pow(max(dot(N, H), 0.0), shininess);
31        vec4 specular = ks * specularProduct;
32
33        if (dot(L, N) < 0.0)    specular = vec4(0.0, 0.0, 0.0, 1.0);
```





scene\_smooth\_dir.html

JS scene\_smooth\_dir.js

&lt;&gt; scene\_smooth\_point.html

JS scene\_smooth\_point.js

&lt;&gt; scene\_smooth\_spot.html

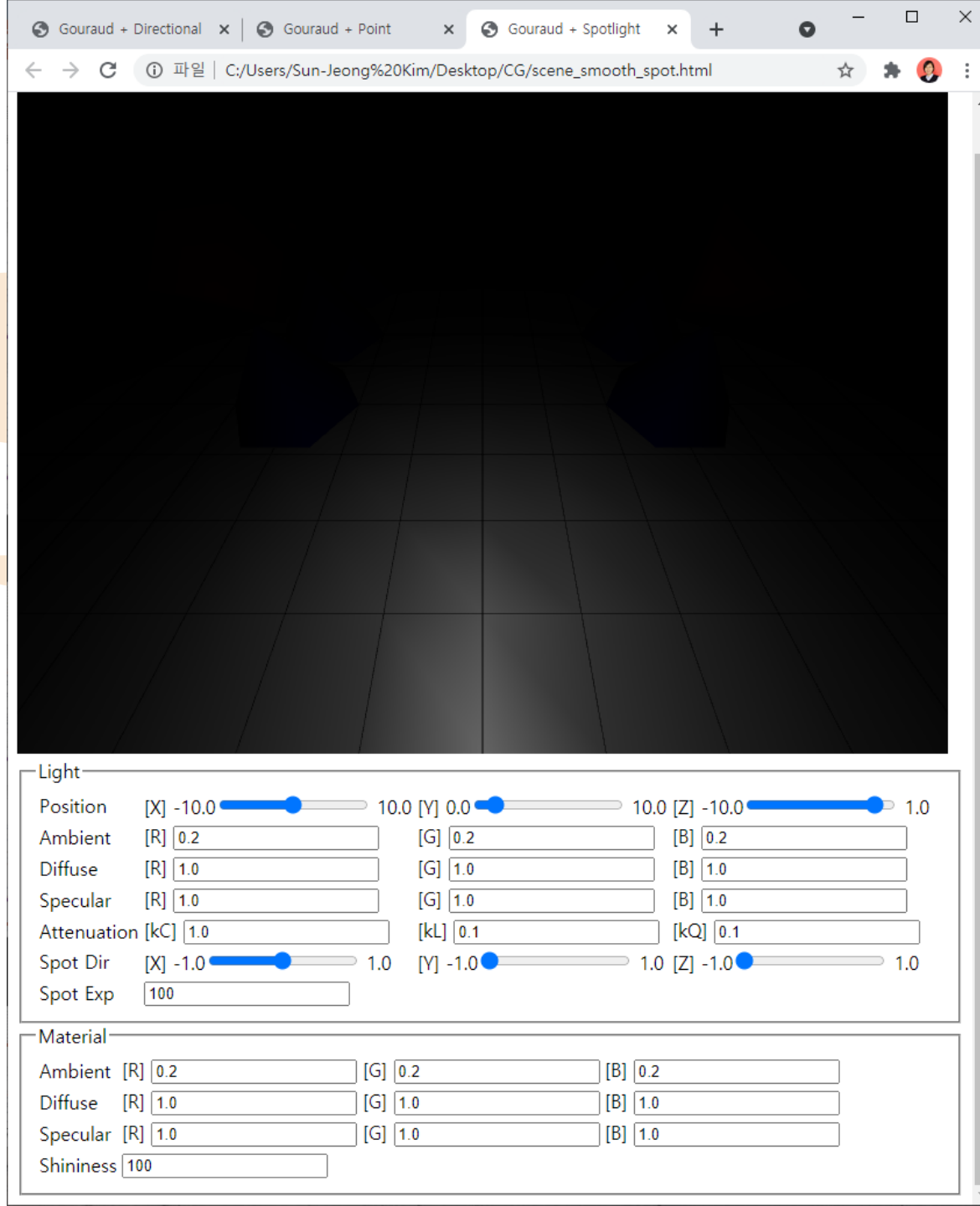
JS scene\_smooth\_spot.js X

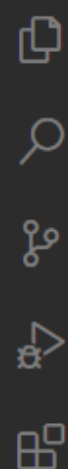


C: &gt; Users &gt; Sun-Jeong Kim &gt; Desktop &gt; CG &gt; JS scene\_smooth\_spot.js &gt; setLighting

```
1  var gl;
2  var points = [];
3  var normals = [];
4
5  var theta = 0.0;
6
7  var viewMatrix;
8  var modelViewMatrixLoc;
9  var lightPosLoc, ambientProductLoc, diffuseProductLoc, specularProductLoc, shininessLoc;
10 var lightAttenLoc, spotDirLoc, spotExpLoc;
11
12 var trballMatrix = mat4(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);
13
14 var numVertCubeTri, numVertPyraTri, numVertGroundTri, numVertGroundLine;
15
16 window.onload = function init()
17 {
18     var canvas = document.getElementById("gl-canvas");
19
20     gl = WebGLUtils.setupWebGL(canvas);
21     if( !gl ) {
22         alert("WebGL isn't available!");
23     }
24
25     generateCube();
26     generateHexaPyramid();
27     generateGround(10);
28
29     // virtual trackball
30     var trball = trackball(canvas.width, canvas.height);
31     var mouseDown = false;
32
33     canvas.addEventListener("mousedown", function (event) {
```







scene\_smooth\_dir.js

&lt;&gt; scene\_smooth\_point.html

JS scene\_smooth\_point.js

&lt;&gt; scene\_smooth\_spot.html

JS scene\_smooth\_spot.js

&lt;&gt; scene\_phong\_dir.html ×



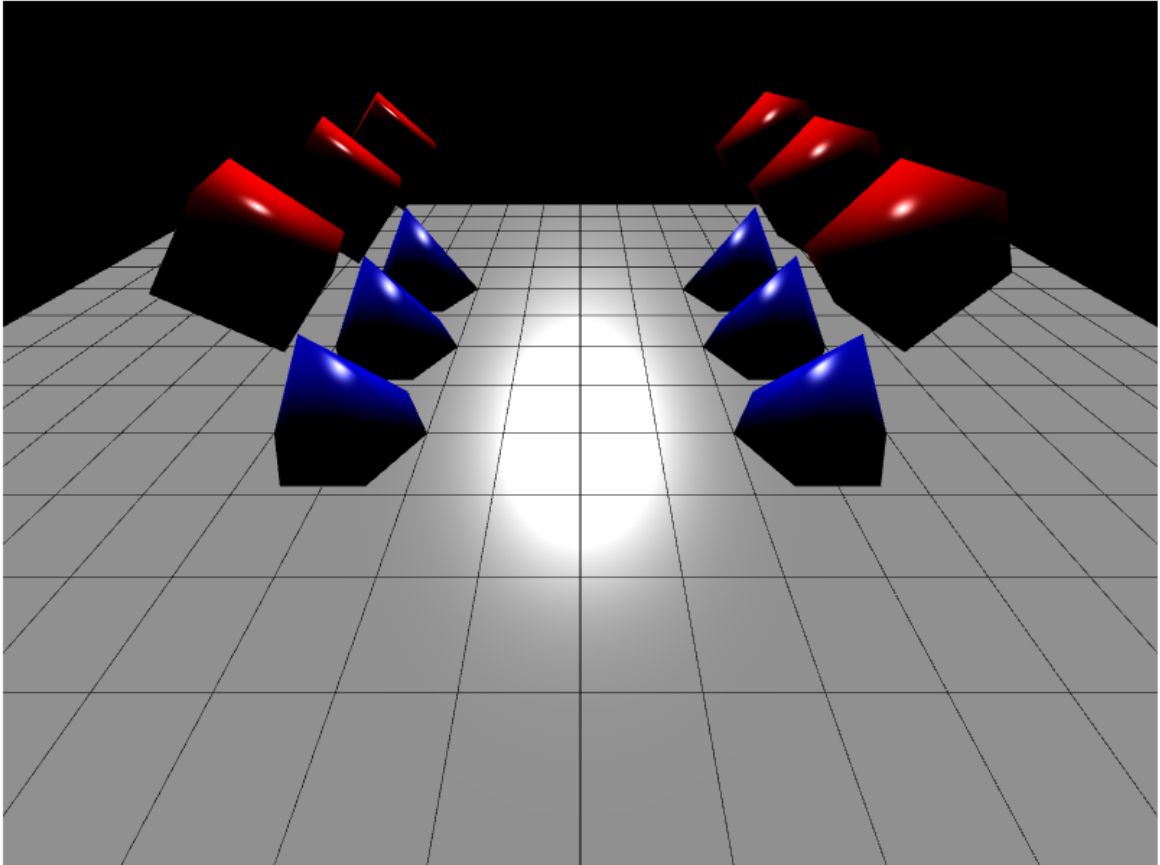
C: &gt; Users &gt; Sun-Jeong Kim &gt; Desktop &gt; CG &gt; &lt;&gt; scene\_phong\_dir.html &gt; html &gt; head &gt; script#fragment-shader

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Phong + Directional</title>
5      <script id="vertex-shader" type="x-shader/x-vertex">
6          attribute vec4 vPosition;
7          attribute vec4 vNormal;
8          uniform mat4 modelViewMatrix;
9          uniform mat4 projectionMatrix;
10
11         varying vec3 fNormal, fPosition;
12
13         void main() {
14             gl_Position = projectionMatrix * modelViewMatrix * vPosition;
15
16             fNormal = normalize((modelViewMatrix * vNormal).xyz);
17             fPosition = (modelViewMatrix * vPosition).xyz;
18         }
19     </script>
20
21     <script id="fragment-shader" type="x-shader/x-fragment">
22         precision mediump float;
23
24         varying vec3 fNormal, fPosition;
25
26         uniform vec4 lightPos, ambientProduct, diffuseProduct, specularProduct;
27         uniform float shininess;
28
29         void main() {
30             vec3 N = normalize(fNormal);
31             vec3 L = normalize(lightPos.xyz);
32             float kd = max(dot(L, N), 0.0);
33             vec4 diffuse = kd * diffuseProduct;
```



Gouraud + Dir x Gouraud + Poi x Gouraud + Sph x Phong + Direc x

파일 C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene\_phong\_dir.html



Light

Direction [X] -1.0 1.0 [Y] 0.0 2.0 [Z] -1.0 1.0

Ambient [R] 0.2 [G] 0.2 [B] 0.2

Diffuse [R] 1.0 [G] 1.0 [B] 1.0

Specular [R] 1.0 [G] 1.0 [B] 1.0

Material

Ambient [R] 0.2 [G] 0.2 [B] 0.2

Diffuse [R] 1.0 [G] 1.0 [B] 1.0

Specular [R] 1.0 [G] 1.0 [B] 1.0

Shininess 100



scene\_smooth\_point.html

JS scene\_smooth\_point.js

&lt;&gt; scene\_smooth\_spot.html

JS scene\_smooth\_spot.js

&lt;&gt; scene\_phong\_dir.html

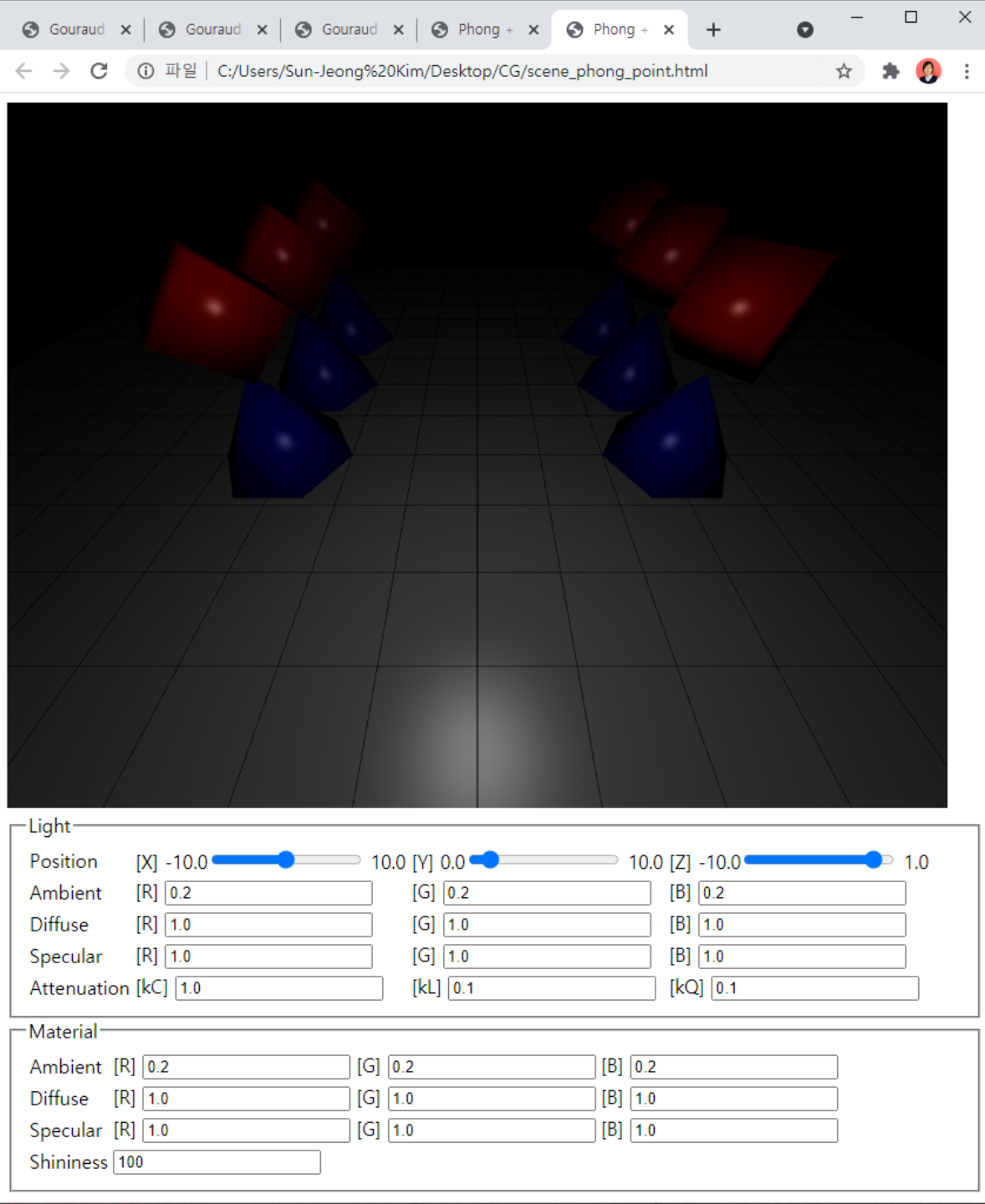
&lt;&gt; scene\_phong\_point.html X



C: &gt; Users &gt; Sun-Jeong Kim &gt; Desktop &gt; CG &gt; &lt;&gt; scene\_phong\_point.html &gt; html &gt; head &gt; title

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Phong + Point</title>
5      <script id="vertex-shader" type="x-shader/x-vertex">
6          attribute vec4 vPosition;
7          attribute vec4 vNormal;
8          uniform mat4 modelViewMatrix;
9          uniform mat4 projectionMatrix;
10
11          varying vec3 fNormal, fPosition;
12
13          void main() {
14              gl_Position = projectionMatrix * modelViewMatrix * vPosition;
15
16              fNormal = normalize((modelViewMatrix * vNormal).xyz);
17              fPosition = (modelViewMatrix * vPosition).xyz;
18          }
19      </script>
20
21      <script id="fragment-shader" type="x-shader/x-fragment">
22          precision mediump float;
23
24          varying vec3 fNormal, fPosition;
25
26          uniform vec4 lightPos, ambientProduct, diffuseProduct, specularProduct;
27          uniform vec3 kAtten;
28          uniform float shininess;
29
30          void main() {
31              vec3 N = normalize(fNormal);
32              vec3 pos = fPosition;
33              vec3 light = lightPos.xyz - pos;
```



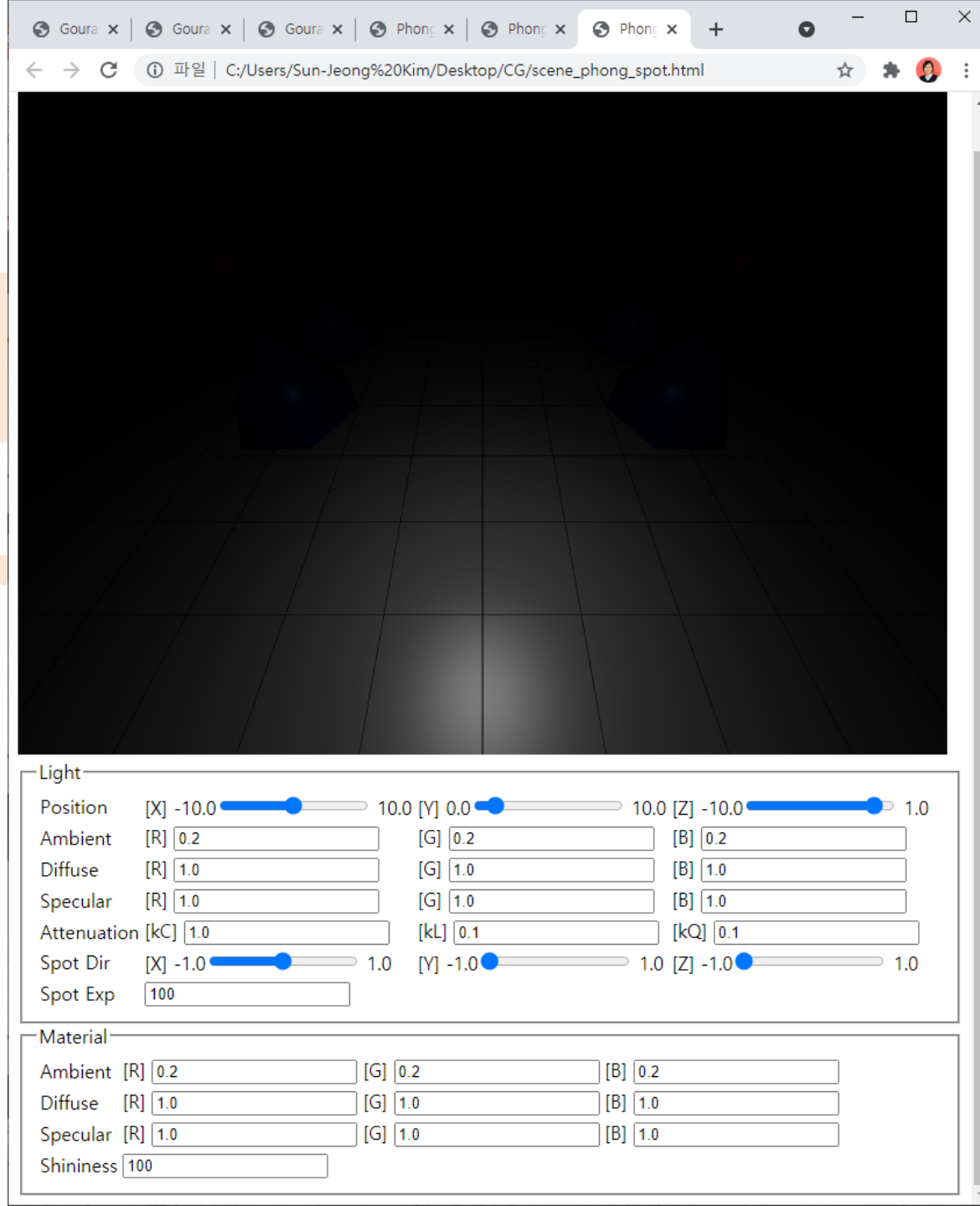


C: &gt; Users &gt; Sun-Jeong Kim &gt; Desktop &gt; CG &gt; &lt;&gt; scene\_phong\_spot.html &gt; html &gt; head &gt; title

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Phong + Spotlight</title>
5      <script id="vertex-shader" type="x-shader/x-vertex">
6          attribute vec4 vPosition;
7          attribute vec4 vNormal;
8          uniform mat4 modelViewMatrix;
9          uniform mat4 projectionMatrix;
10
11         varying vec3 fNormal, fPosition;
12
13         void main() {
14             gl_Position = projectionMatrix * modelViewMatrix * vPosition;
15
16             fNormal = normalize((modelViewMatrix * vNormal).xyz);
17             fPosition = (modelViewMatrix * vPosition).xyz;
18         }
19     </script>
20
21     <script id="fragment-shader" type="x-shader/x-fragment">
22         precision mediump float;
23
24         varying vec3 fNormal, fPosition;
25
26         uniform vec4 lightPos, ambientProduct, diffuseProduct, specularProduct;
27         uniform vec3 kAtten, spotDir;
28         uniform float shininess, spotExp;
29
30         void main() {
31             vec3 N = normalize(fNormal);
32             vec3 pos = fPosition;
33             vec3 light = lightPos.xyz - pos;
```







# 연습 문제

- $(0, 1, 0)$  위치에 Sphere를 위치 시키시오.
  - Subdivision Level은 4 또는 5로 고정
  - 법선 벡터는 좌표와 동일

