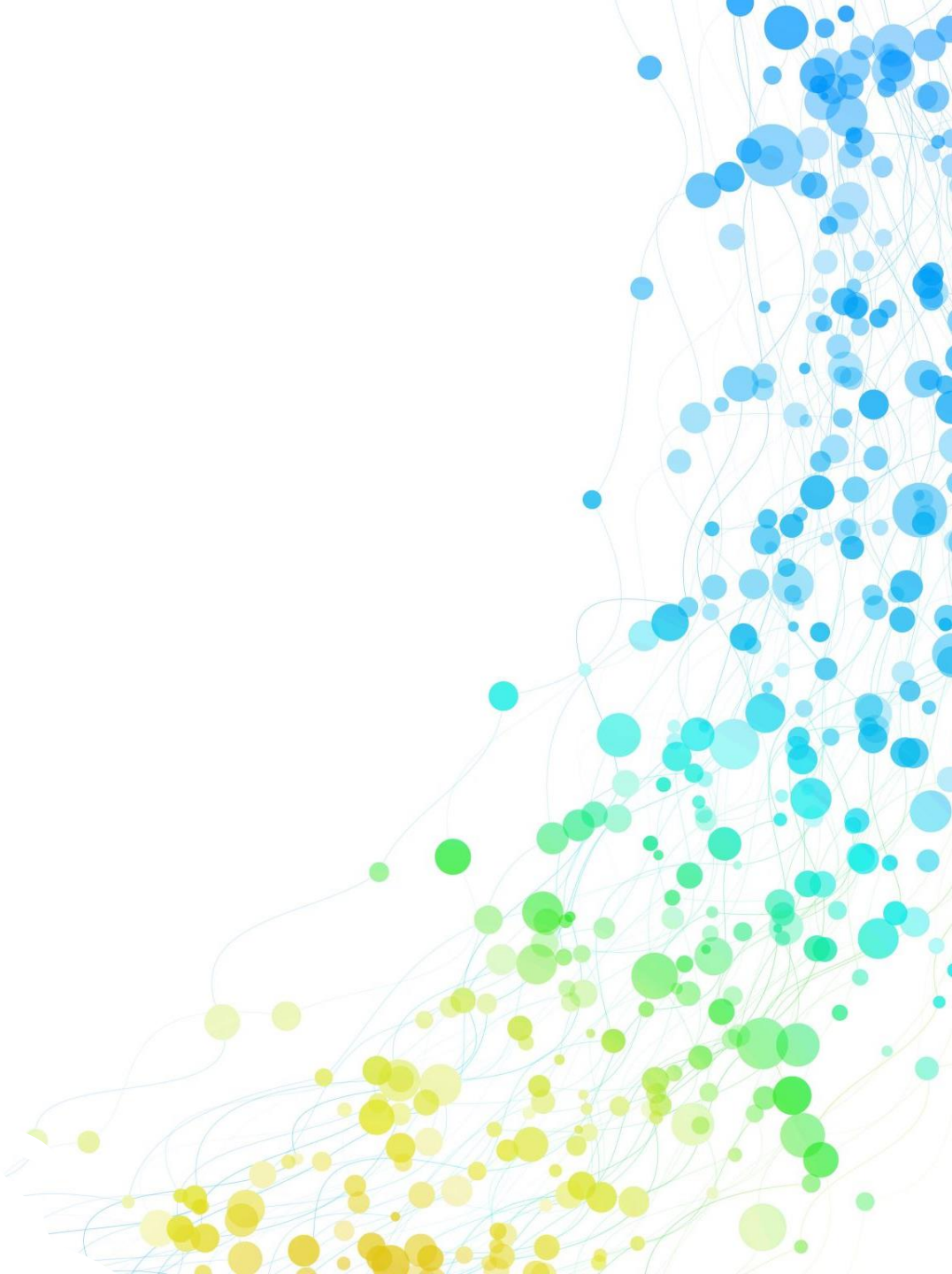
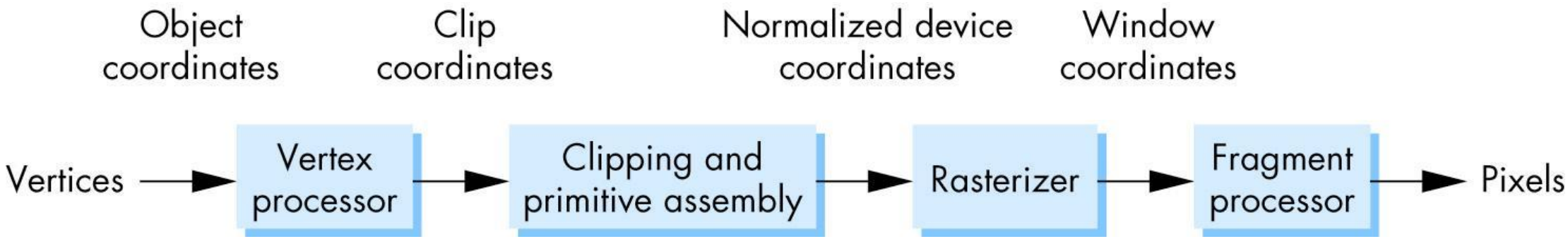
Graphics Programming

5TH WEEK, 2022



Programmable Pipelines

- Two components
  - Vertex program (vertex shaders)
  - Fragment program (fragment shaders)
- In the pipeline architecture, the vertex processor and the fragment processor are programmable by application programs called shaders



Vertex Shader Applications

- Moving vertices
  - Morphing
  - Wave motion
  - Fractals

- Lighting
  - More realistic models
  - Cartoon shaders

Fragment Shader Applications (1)

- Per fragment lighting calculations

Per vertex lighting

Per fragment lighting

Fragment Shader Applications (2)

- Texture mapping

Smooth shading

Environment mapping

환경매핑

Bump mapping

Simple Vertex Shader

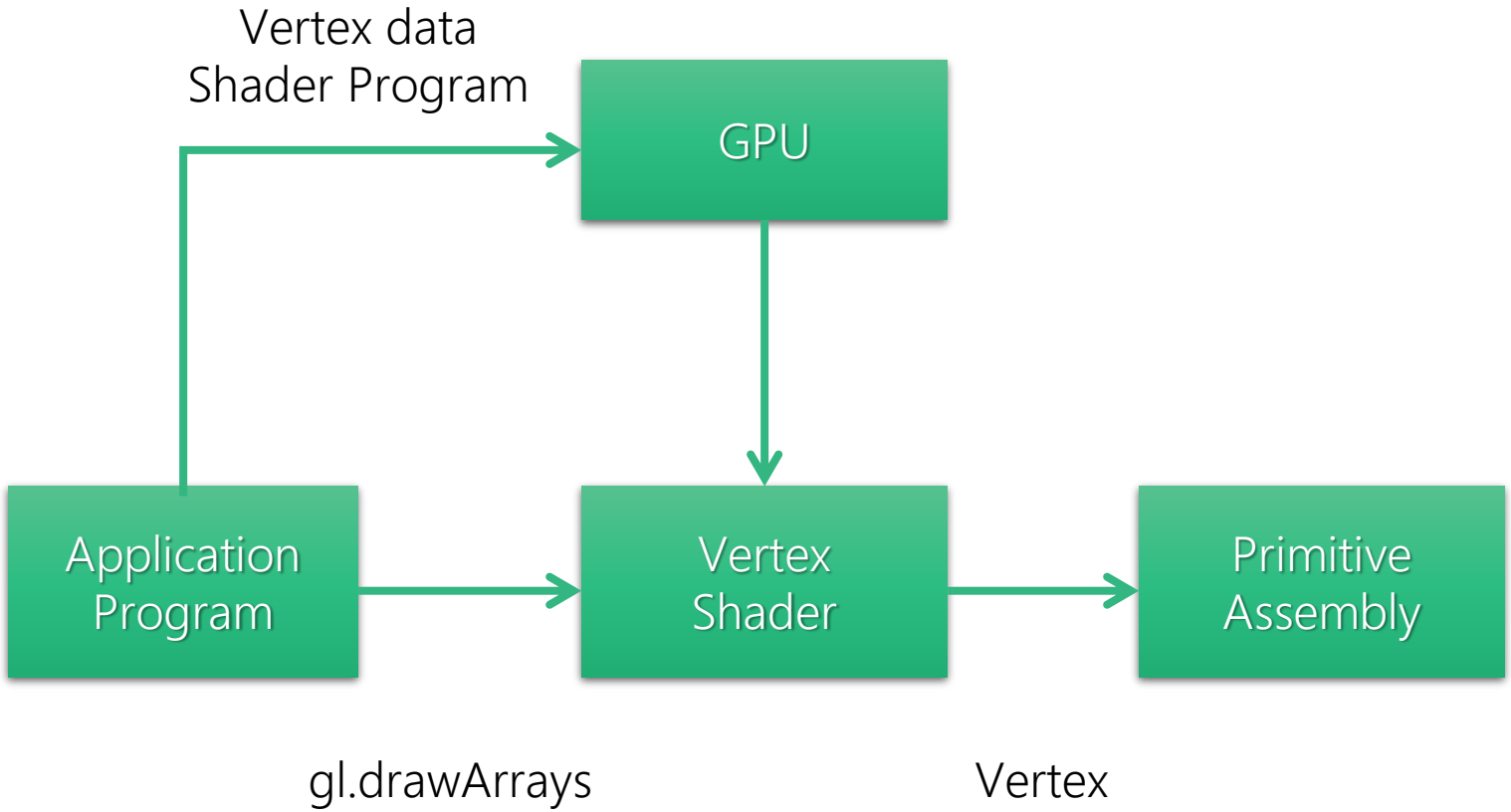
input from application

```
attribute vec4 vPosition;

void main(void)
{
    gl_Position = vPosition;
}
```

must link to variable in application

built in variable

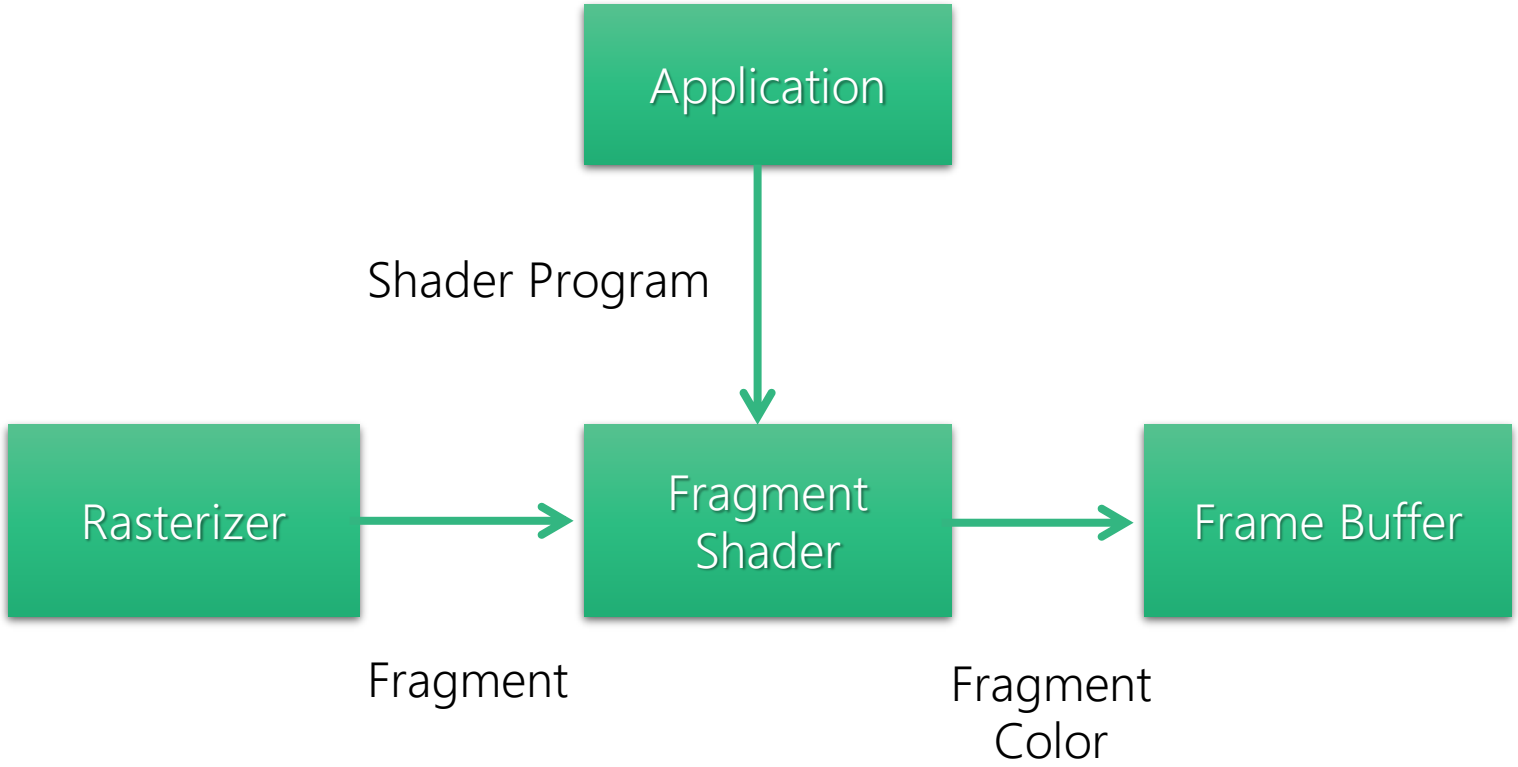
Execution Model – Vertex Shader



Simple Fragment Shader

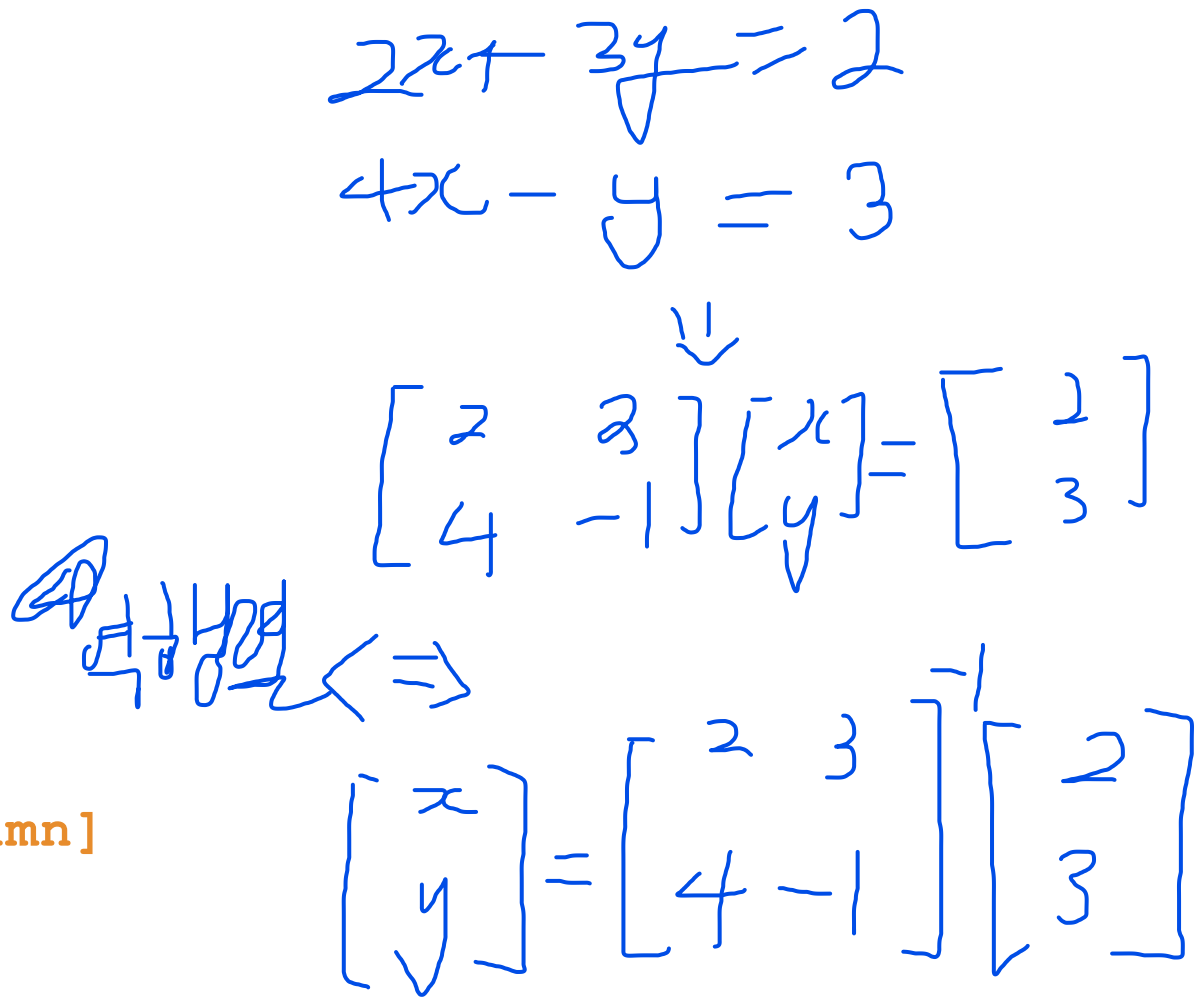
```
precision mediump float;

void main(void)
{
  gl_FragColor = vec4( 1.0, 0.0, 0.0, 1.0 );
}
```

Execution Model – Fragment Shader



Data Type

- C types: `int`, `float`, `bool`
- Vectors:
  - float `vec2`, `vec3`, `vec4`
  - Also int (`ivec`) and boolean (`bvec`)
- Matrices: `mat2`, `mat3`, `mat4`
  - Stored by columns
  - Standard referencing `m[row][column]`
- C++ style constructors
  - `vec3 a = vec3(1.0, 2.0, 3.0);`
  - `vec2 b = vec2(a);`



Pointers

- There are no pointers in GLSL
- We can use C structs which can be copied back from functions
- Because matrices and vectors are basic types they can be passed into and output from GLSL functions
  - Ex) `mat3 func(mat3 a);`
- Variables passed by copying

Qualifiers

- GLSL has many of the same qualifiers such as `const` as C/C++
- Need others due to the nature of the shader architecture
- Variables can change
  - Once per primitive
  - Once per vertex
  - Once per fragment
  - At any time in the application
- Vertex attributes are interpolated by the rasterizer into fragment attributes

Attribute Qualifiers

- Attribute-qualified variables can change at most once per vertex
- There are a few built in variables such as `gl_Position` but most have been deprecated
- User defined (in application program)
  - `attribute float temperature;`
  - `attribute vec3 velocity;`
  - Recent versions of GLSL use `in` and `out` qualifiers to get to and from shaders

Uniform Qualifiers

- Variables that are constant for an entire primitive
- Can be changed in application and sent to shaders
- Cannot be changed in shader
- Used to pass information to shader such as the bounding box of a primitive

Varying Qualifiers

- Variables that are passed from vertex shader to fragment shader
- Automatically interpolated by the rasterizer
- With WebGL, GLSL uses the varying qualifier in both shaders
  - `varying vec4 color;`
- More recent versions of WebGL use `out` in vertex shader and `in` in the fragment shader
  - `out vec4 color;  // vertex shader`
  - `in vec4 color;  // fragment shader`

Example: Vertex Shader

```
attribute vec4 vPosition;
attribute vec4 vColor;
varying vec4 fColor;
void main(void)
{
    gl_Position = vPosition;
    fColor = vColor;
}
```

Corresponding Fragment Shader

```
precision mediump float;

varying vec4 fColor;
void main(void)
{
   gl_FragColor = fColor;
}
```

Sending Colors from Application

```
var cBuffer = gl.createBuffer();

gl.bindBuffer(gl.ARRAY_BUFFER, cBuffer);

gl.bufferData(gl.ARRAY_BUFFER, flatten(colors),
gl.STATIC_DRAW);


var vColor = gl.getAttribLocation(program, "vColor");

gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 0);

gl.enableVertexAttribArray(vColor);
```

Sending a Uniform Variable

```
// in application
vec4 color = vec4(1.0, 0.0, 0.0, 1.0);
colorLoc = gl.getUniformLocation(program, ”uColor");
gl.uniform4f(colorLoc, color);

// in fragment shader (similar in vertex shader)
uniform vec4 uColor;
void main(){
    gl_FragColor = uColor;
}
```

Operators and Functions

- Standard C functions
  - Trigonometric
  - Arithmetic
  - Normalize, reflect, length

- Overloading of vector and matrix types

```
mat4 a;
vec4 b, c, d;
c = b*a; // a column vector stored as a 1d array
d = a*b; // a row vector stored as a 1d array
```

Swizzling and Selection

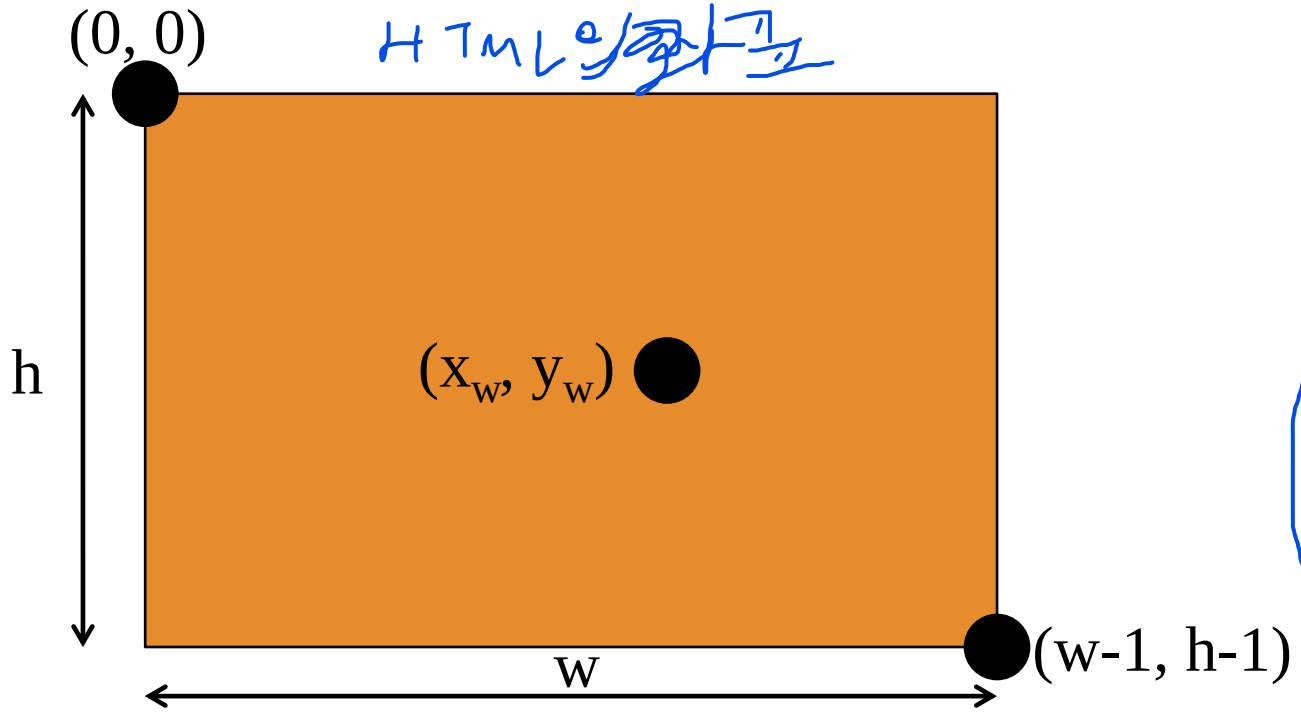
- Can refer to array elements by element using [ ] or selection (.) operator with
  - `x`, `y`, `z`, `w`
  - `r`, `g`, `b`, `a`
  - `s`, `t`, `p`, `q`
  - Ex) `a[2]`, `a.b`, `a.z`, `a.p` are the same

- Swizzling operator lets us manipulate components

```
vec4 a = vec4(1.0, 2.0, 3.0, 4.0);
a.yz = vec2(1.0, 2.0); // a = (1.0, 1.0, 2.0, 4.0)
```

Position Input

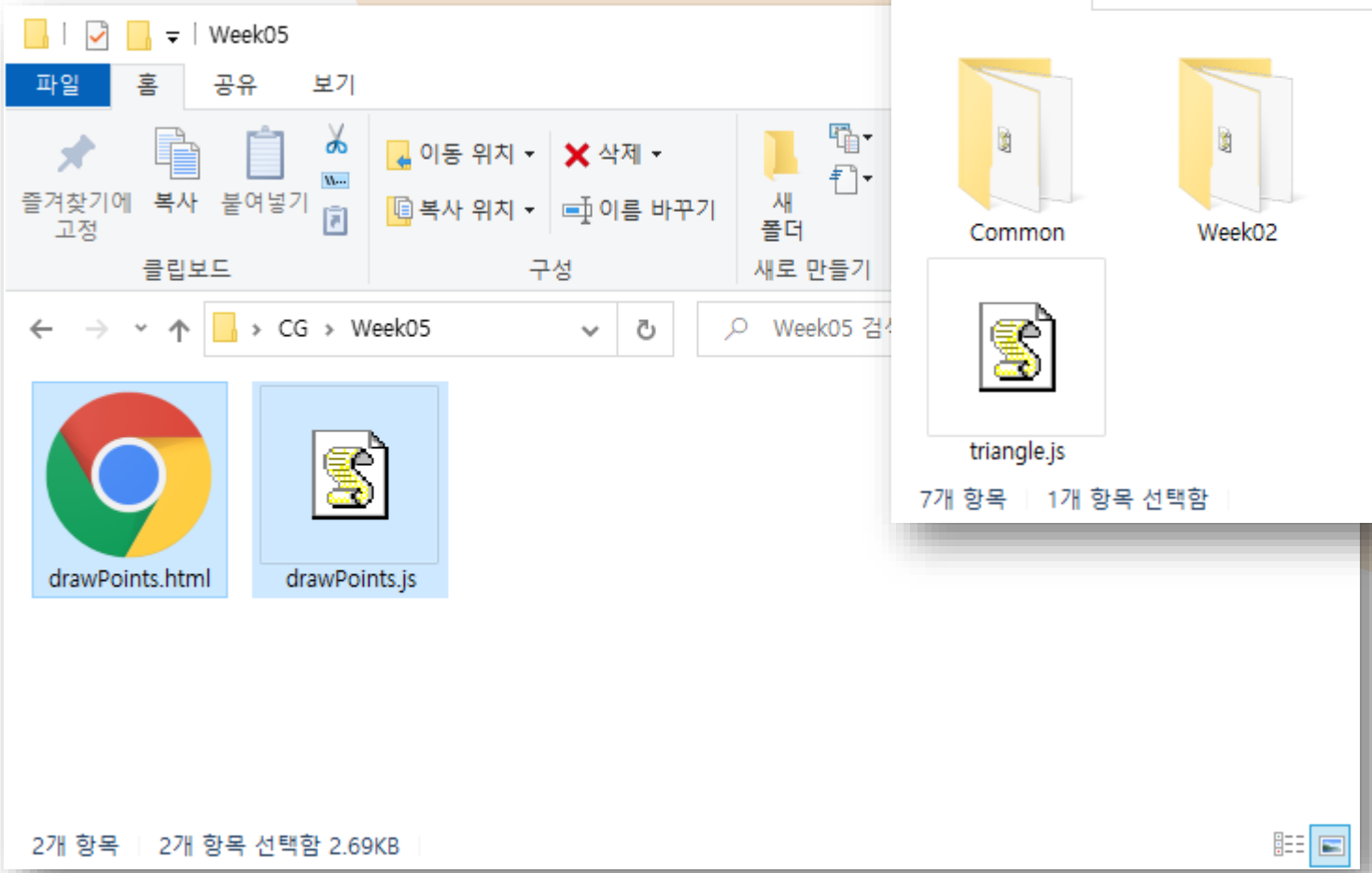
- Returning position from click event
  - Canvas specified in HTML file of size canvas.width x canvas.height
  - Returned window coordinates are event.clientX and event.clientY

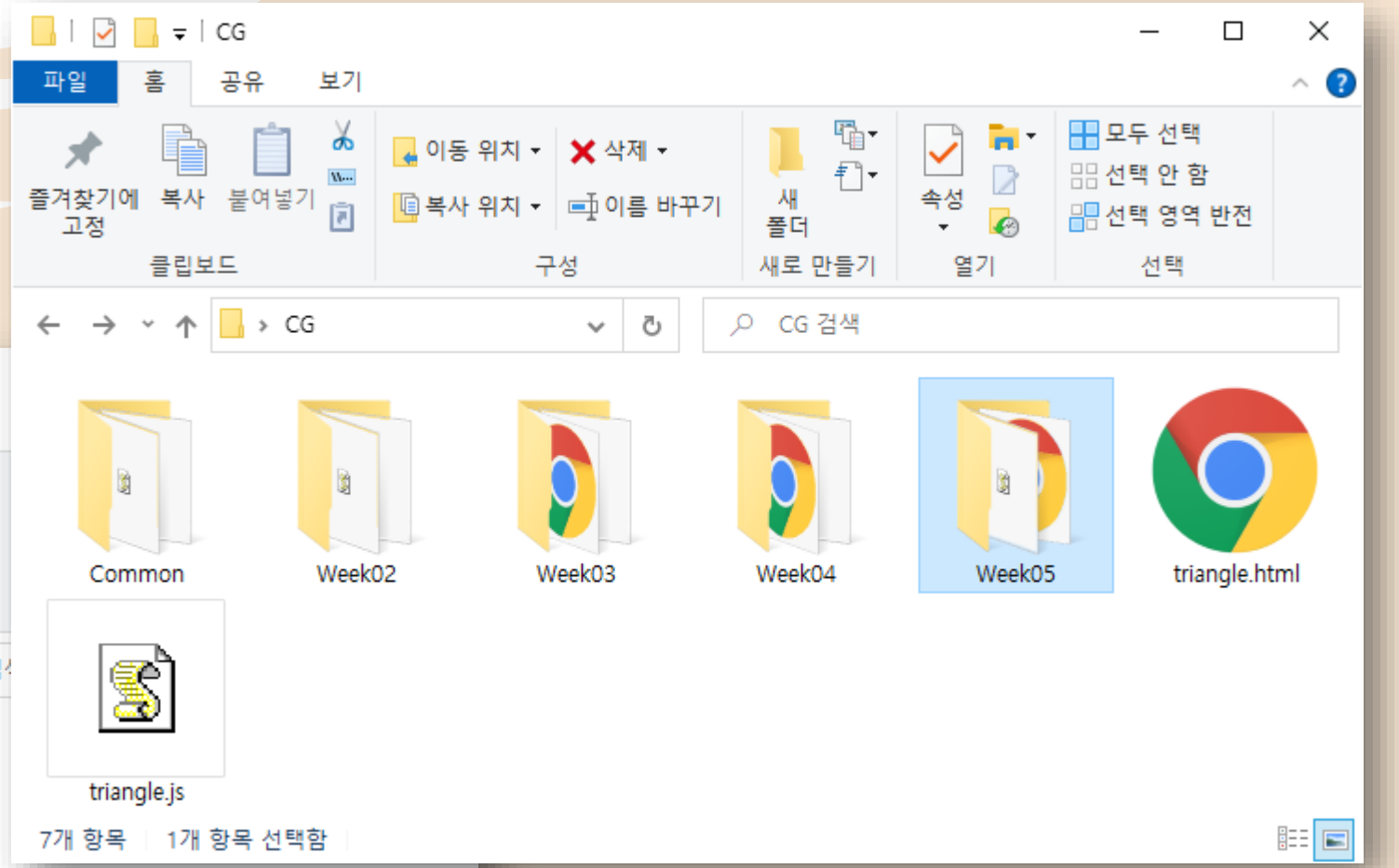


$$(0,h) \rightarrow (-1,-1)$$

$$(w,0) \rightarrow (1,1)$$

$$\begin{cases} x = -1 + \dfrac{2 * x_w}{w} \\ y = -1 + \dfrac{2 * (h - y_w)}{h} \end{cases}$$


Week05
파일
홈
공유
보기
즐겨찾기에 고정
복사
붙여넣기
이동 위치
복사 위치
삭제
이름 바꾸기
새 폴더
클립보드
구성
새로 만들기
CG
Week05
drawPoints.html
drawPoints.js
2개 항목
2개 항목 선택함 2.69KB


CG
파일
홈
공유
보기
즐겨찾기에 고정
복사
붙여넣기
이동 위치
복사 위치
삭제
이름 바꾸기
새 폴더
속성
모두 선택
선택 안 함
선택 영역 반전
클립보드
구성
새로 만들기
열기
선택
CG 검색
Common
Week02
Week03
Week04
Week05
triangle.html
triangle.js
7개 항목
1개 항목 선택함

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

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

drawPoints.html drawPoints.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawPoints.html > ...

```
<!DOCTYPE html>
<html>
    <head>
        <title>2022 Computer Graphics - Drawing Points</title>

        <script id="vertex-shader" type="x-shader/x-vertex">
        attribute vec4 vPosition;

        void main() {
            gl_PointSize = 5.0;
            gl_Position = vPosition;
        }
        </script>

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

        void main() {
            gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
        }
        </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="drawPoints.js"></script>
    </head>
    <body>
        <canvas id="gl-canvas" width="512" height="512">
            Oops... your browser doesn't support the HTML5 canvas element!
        </canvas>
    </body>
</html>
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 1, Col 1 Spaces: 4 UTF-8 CRLF HTML

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

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

drawPoints.html drawPoints.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawPoints.js > ...

```
1   var gl;
2   var points;
3
4   window.onload = function init() {
5       var canvas = document.getElementById("gl-canvas");
6
7       gl = WebGLUtils.setupWebGL(canvas);
8       if( !gl ) {
9           alert("WebGL isn't available!");
10      }
11
12      points = [];
13      var bMouseDown = false;
14
15      canvas.addEventListener("mousedown", function(event) {
16          bMouseDown = true;
17      });
18      canvas.addEventListener("mouseup", function(event) {
19          bMouseDown = false;
20      });
21      canvas.addEventListener("mousemove", function(event) {
22          if( bMouseDown ) {
23              var point = vec2(2 * event.clientX/canvas.width - 1,
24                  2 * (canvas.height - event.clientY) / canvas.height - 1);
25              points.push(point);
26              gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
27
28              render();
29          }
30      });
31
32      // Configure WebGL
33      gl.viewport(0, 0, canvas.width, canvas.height);
34      gl.clearColor(1.0, 1.0, 1.0, 1.0);
35
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 1, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

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

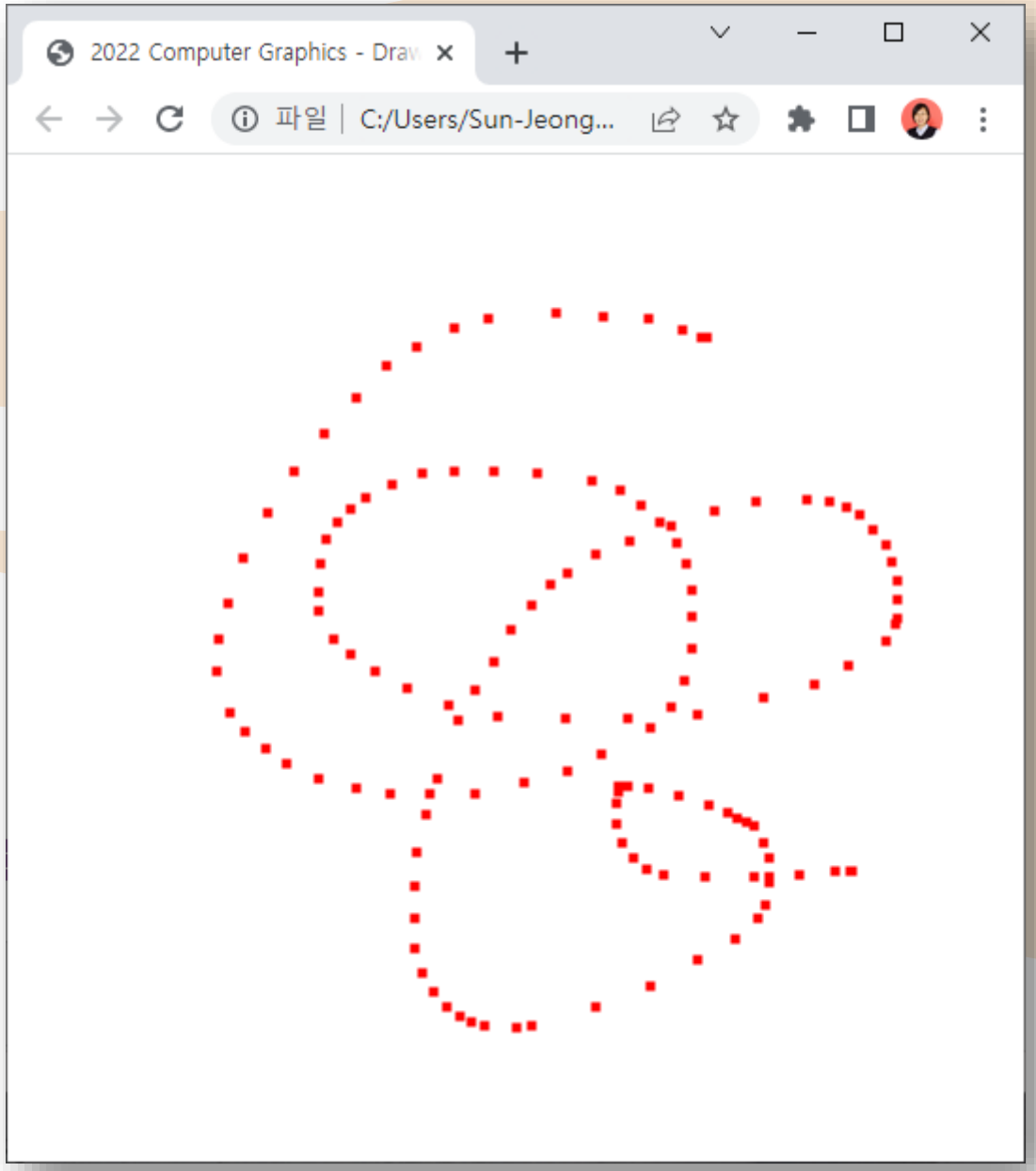
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

drawPoints.html drawPoints.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawPoints.js > ...

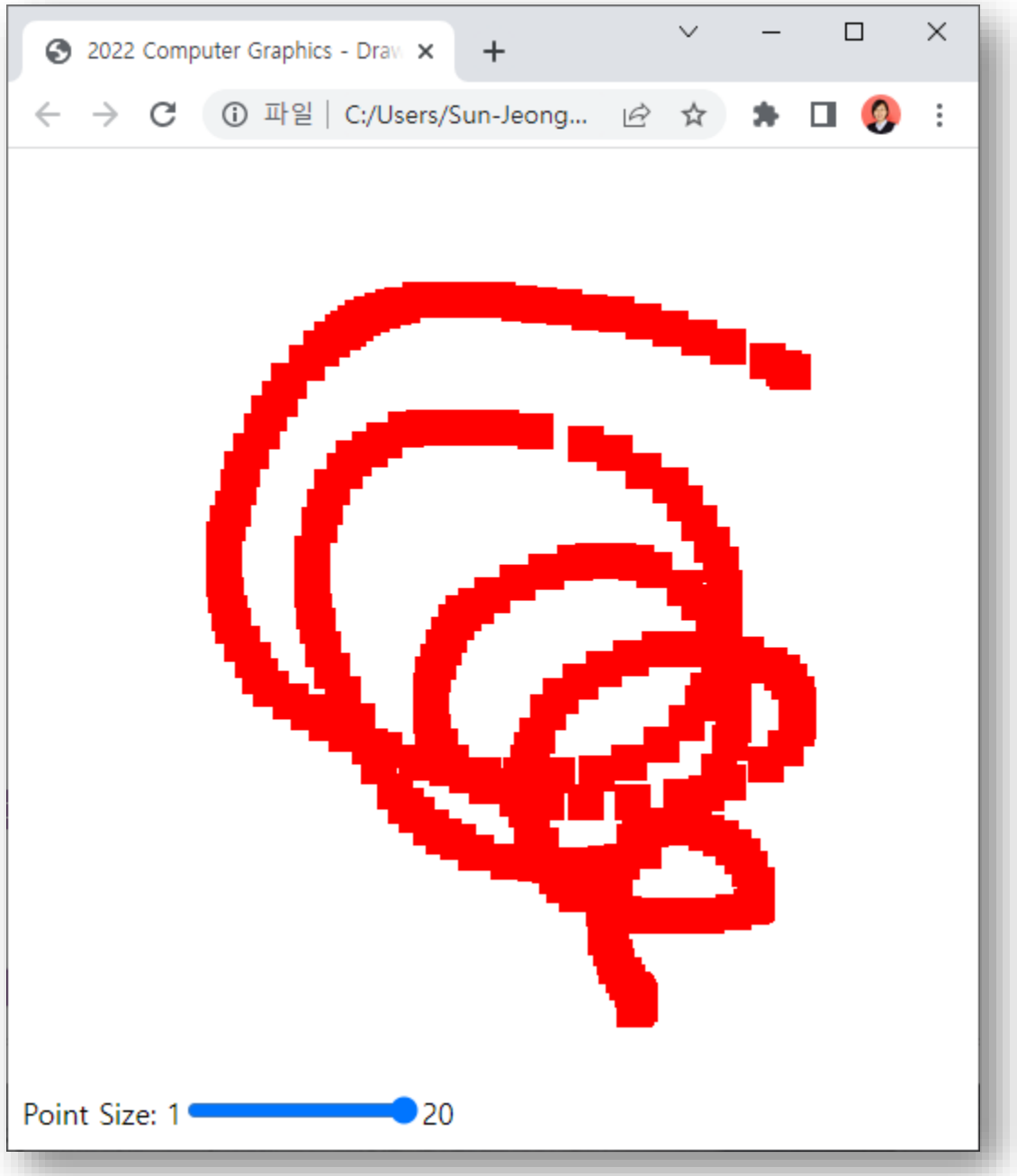
```
35
36      // Load shaders and initialize attribute buffers
37      var program = initShaders(gl, "vertex-shader", "fragment-shader");
38      gl.useProgram(program);
39
40      // Load the data into the GPU
41      var bufferId = gl.createBuffer();
42      gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
43      gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
44
45      // Associate our shader variables with our data buffer
46      var vPosition = gl.getAttribLocation(program, "vPosition");
47      gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
48      gl.enableVertexAttribArray(vPosition);
49
50      render();
51  };
52
53  function render() {
54      gl.clear(gl.COLOR_BUFFER_BIT);
55      gl.drawArrays(gl.POINTS, 0, points.length);
56  }
57
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 1, Col 1 Spaces: 4 UTF-8 CRLF JavaScript


2022 Computer Graphics - Draw
파일 | C:/Users/Sun-Jeong...

연습 문제 (1)

- 웹페이지에 슬라이더를 추가하여, 점의 크기를 입력 받아 그리시오.



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

drawPoints.html drawPoints.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawPoints.html > html > body > div

```
<head>
    <title>2022 Computer Graphics - Drawing Points</title>

    <script id="vertex-shader" type="x-shader/x-vertex">
    attribute vec4 vPosition;
    uniform float pointSize;

    void main() {
        gl_PointSize = pointSize;
        gl_Position = vPosition;
    }
    </script>

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

    void main() {
        gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
    }
    </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="drawPoints.js"></script>
</head>
<body>
    <canvas id="gl-canvas" width="512" height="512">
        Oops... your browser doesn't support the HTML5 canvas element!
    </canvas>
    <div>
        Point Size: 1<input type="range" id="pointSize" min="1" max="20" step="1" value="10">20
    </div>
</body>
</html>
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 34, Col 100 Spaces: 4 UTF-8 CRLF HTML

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

drawPoints.html drawPoints.js

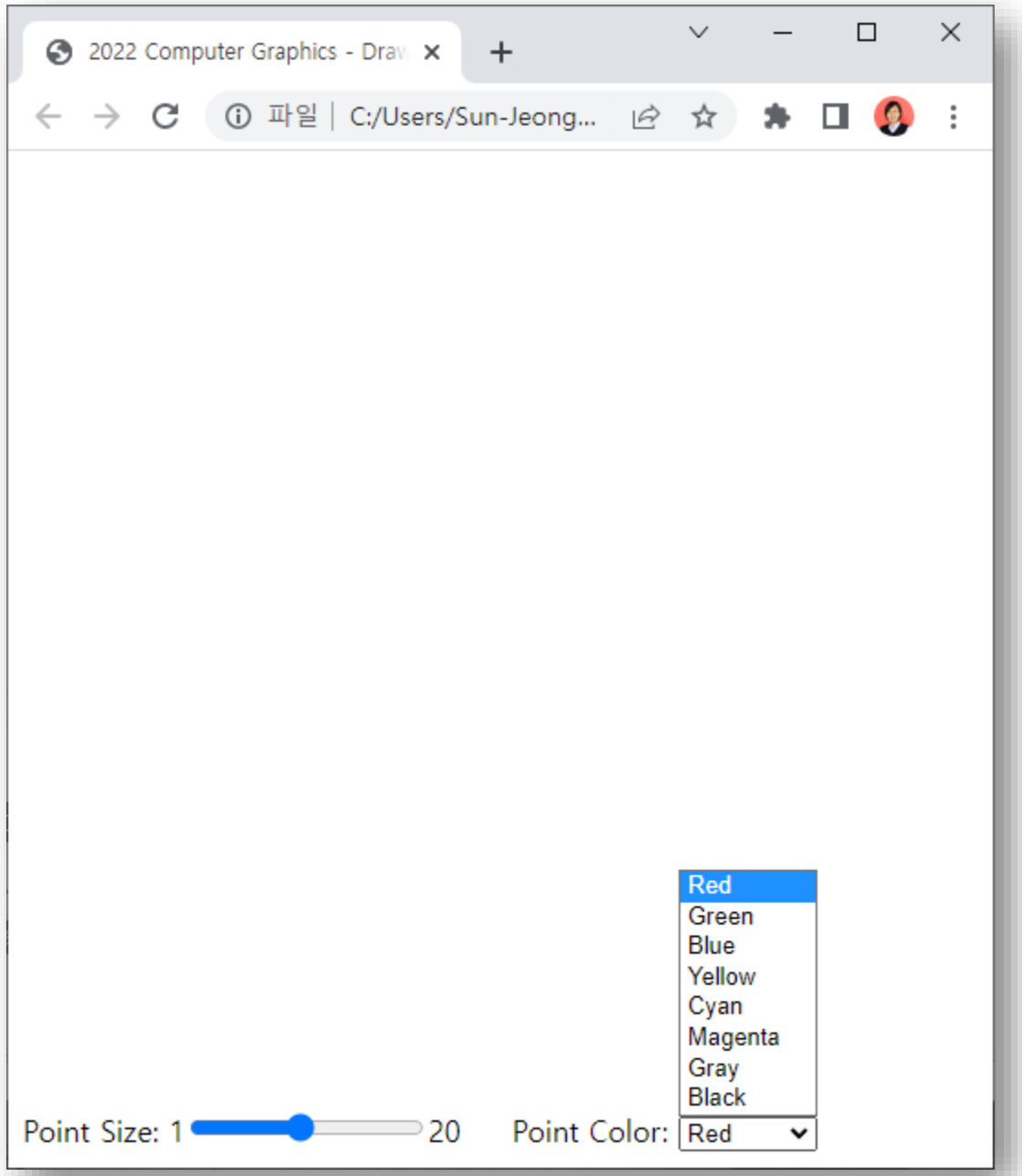
C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawPoints.js > init > onchange

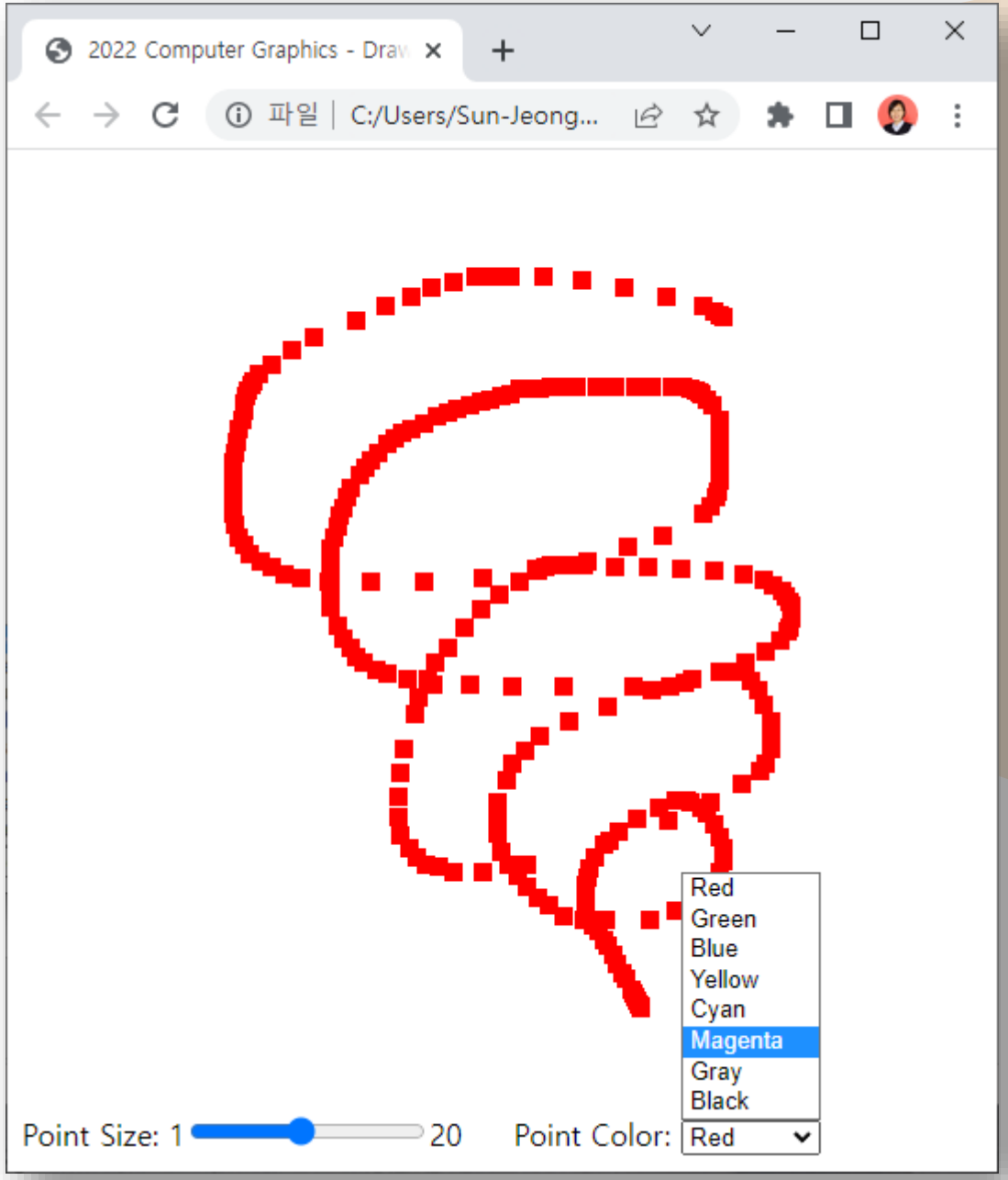
```
35
36      // Load shaders and initialize attribute buffers
37      var program = initShaders(gl, "vertex-shader", "fragment-shader");
38      gl.useProgram(program);
39
40      // Load the data into the GPU
41      var bufferId = gl.createBuffer();
42      gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
43      gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
44
45      // Associate our shader variables with our data buffer
46      var vPosition = gl.getAttribLocation(program, "vPosition");
47      gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
48      gl.enableVertexAttribArray(vPosition);
49
50      var pointSize = gl.getUniformLocation(program, "pointSize");
51      gl.uniform1f(pointSize, 10.0);
52
53      document.getElementById("pointSize").onchange = function() {
54          var size = this.value;
55          gl.uniform1f(pointSize, size);
56
57          render();
58      }
59
60      render();
61  };
62
63  function render() {
64      gl.clear(gl.COLOR_BUFFER_BIT);
65      gl.drawArrays(gl.POINTS, 0, points.length);
66  }
67
```

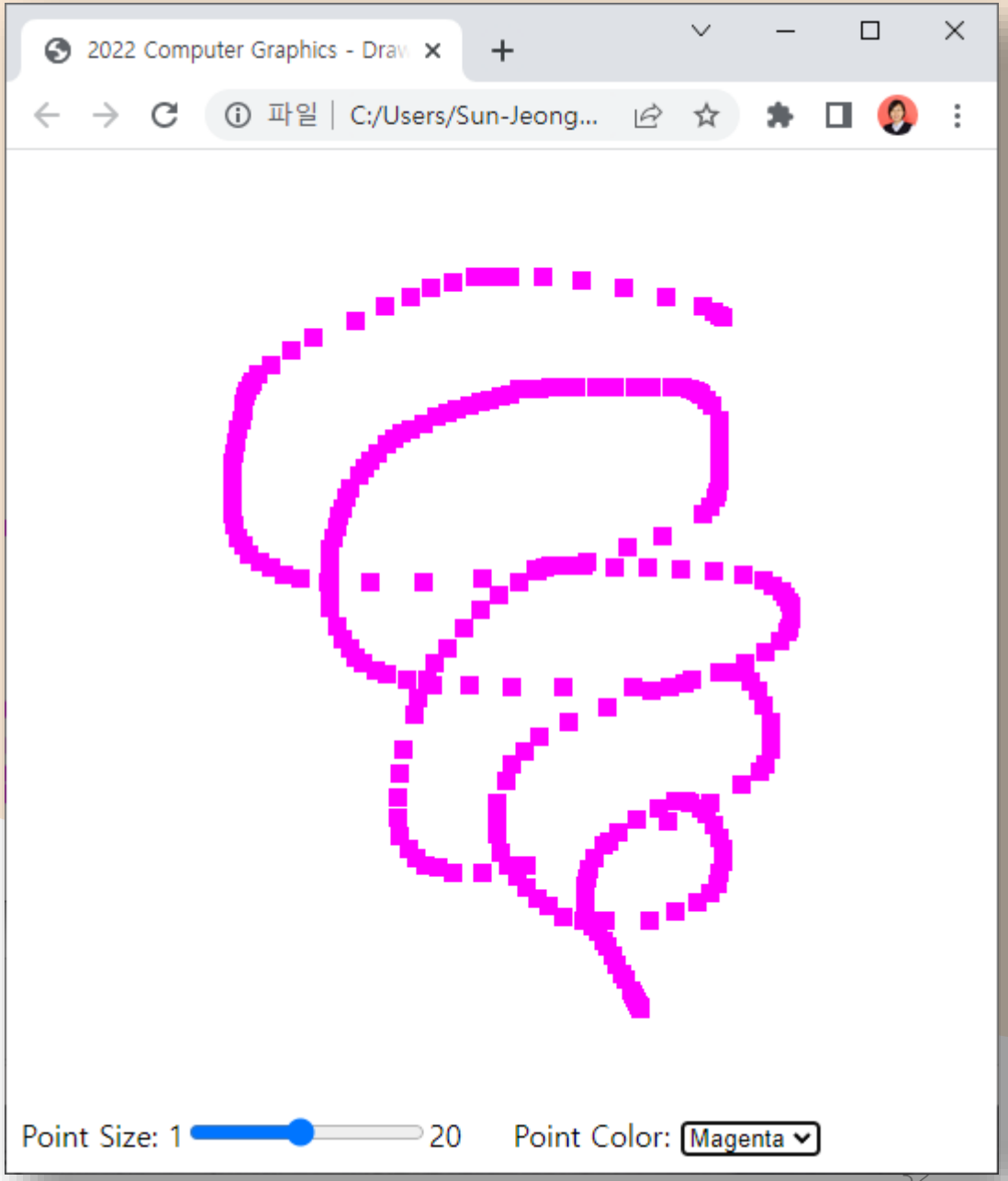
Restricted Mode ⊗ 0 ⚠ 0 Ln 57, Col 18 Spaces: 4 UTF-8 CRLF JavaScript

연습 문제 (2)

- HTML의 Select/Option element를 이용하여 점의 색상을 변경하시오.




2022 Computer Graphics - Draw
파일 | C:/Users/Sun-Jeong...
Red
Green
Blue
Yellow
Cyan
Magenta
Gray
Black
Point Size: 1 20
Point Color: Red


2022 Computer Graphics - Draw
파일 | C:/Users/Sun-Jeong...
Point Size: 1 20
Point Color: Magenta

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

drawPoints.html drawPoints.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawPoints.html > html > body > div > select#pointColor > option

```
15
16          <script id="fragment-shader" type="x-shader/x-fragment">
17          precision mediump float;
18          uniform vec4 fColor;
19
20          void main() {
21              gl_FragColor = fColor;
22          }
23          </script>
24
25          <script type="text/javascript" src="../Common/webgl-utils.js"></script>
26          <script type="text/javascript" src="../Common/initShaders.js"></script>
27          <script type="text/javascript" src="../Common/MV.js"></script>
28          <script type="text/javascript" src="drawPoints.js"></script>
29      </head>
30      <body>
31          <canvas id="gl-canvas" width="512" height="512">
32              Oops... your browser doesn't support the HTML5 canvas element!
33          </canvas>
34          <div>
35              Point Size: 1<input type="range" id="pointSize" min="1" max="20" step="1" value="10">20    
36              Point Color: <select id="pointColor">
37                  <option value="red">Red</option>
38                  <option value="green">Green</option>
39                  <option value="blue">Blue</option>
40                  <option value="yellow">Yellow</option>
41                  <option value="cyan">Cyan</option>
42                  <option value="magenta">Magenta</option>
43                  <option value="gray">Gray</option>
44                  <option value="black">Black</option>
45              </select>
46          </div>
47      </body>
48  </html>
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 44, Col 44 Spaces: 4 UTF-8 CRLF HTML

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

drawPoints.html drawPoints.js

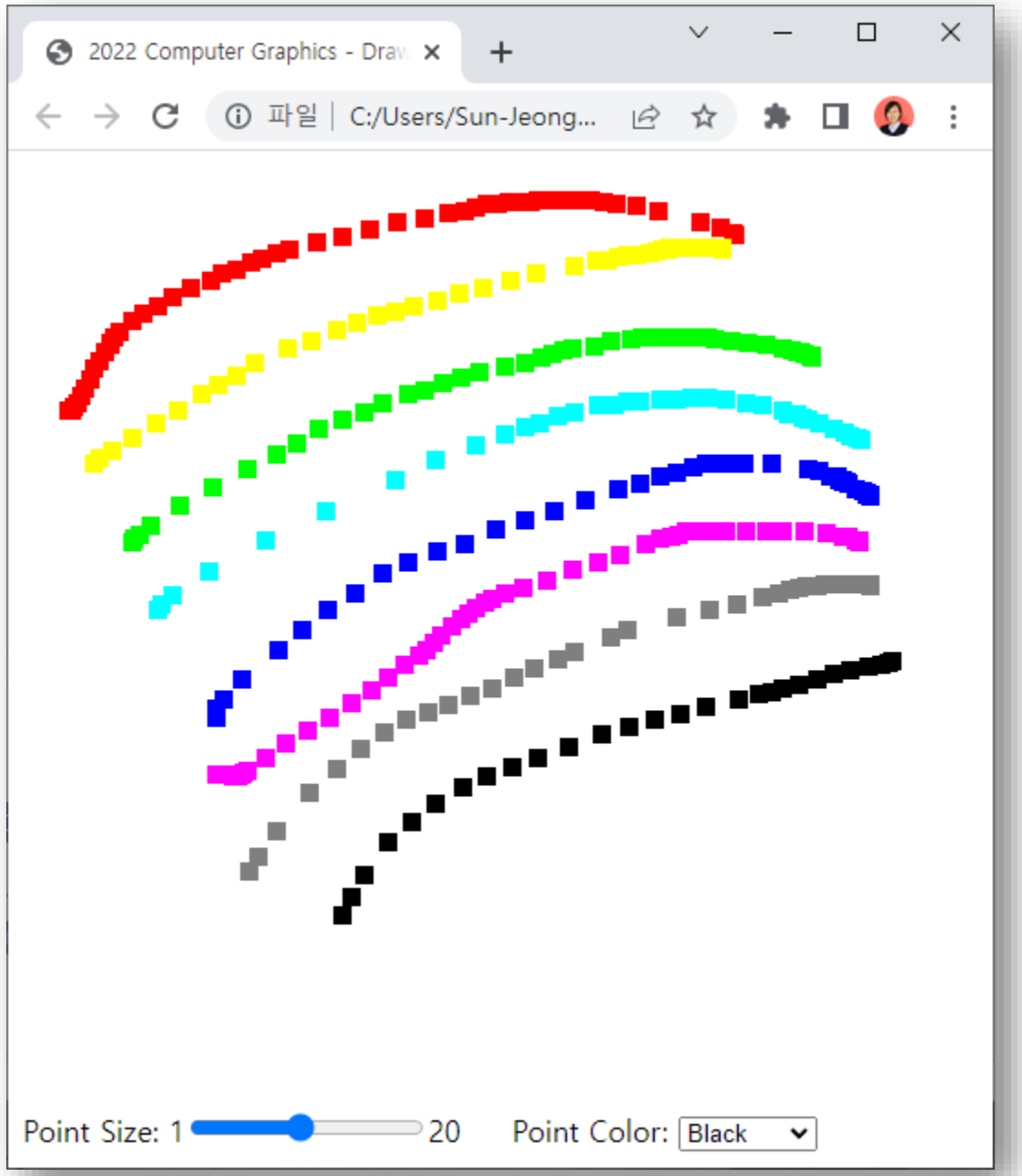
C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawPoints.js > init > onclick

```
59
60      var fColor = gl.getUniformLocation(program, "fColor");
61      gl.uniform4f(fColor, 1.0, 0.0, 0.0, 1.0);
62
63      document.getElementById("pointColor").onclick = function(event) {
64          switch(event.target.value) {
65              case "red":
66                  gl.uniform4f(fColor, 1.0, 0.0, 0.0, 1.0);
67                  break;
68              case "green":
69                  gl.uniform4f(fColor, 0.0, 1.0, 0.0, 1.0);
70                  break;
71              case "blue":
72                  gl.uniform4f(fColor, 0.0, 0.0, 1.0, 1.0);
73                  break;
74              case "yellow":
75                  gl.uniform4f(fColor, 1.0, 1.0, 0.0, 1.0);
76                  break;
77              case "cyan":
78                  gl.uniform4f(fColor, 0.0, 1.0, 1.0, 1.0);
79                  break;
80              case "magenta":
81                  gl.uniform4f(fColor, 1.0, 0.0, 1.0, 1.0);
82                  break;
83              case "gray":
84                  gl.uniform4f(fColor, 0.5, 0.5, 0.5, 1.0);
85                  break;
86              case "black":
87                  gl.uniform4f(fColor, 0.0, 0.0, 0.0, 1.0);
88                  break;
89          }
90
91          render();
92      }
93
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 91, Col 18 Spaces: 4 UTF-8 CRLF JavaScript

연습 문제 (3)

- 점마다 색상을 다르게 채색하시오.



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

drawPoints.html drawPoints.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawPoints.html > html > head > script#fragment-shader

```
<!DOCTYPE html>
<html>
    <head>
        <title>2022 Computer Graphics - Drawing Points</title>

        <script id="vertex-shader" type="x-shader/x-vertex">
        attribute vec4 vPosition;
        attribute vec4 vColor;
        uniform float pointSize;
        varying vec4 fColor;

        void main() {
            fColor = vColor;
            gl_PointSize = pointSize;
            gl_Position = vPosition;
        }
        </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="drawPoints.js"></script>
    </head>
    <body>
        <canvas id="gl-canvas" width="512" height="512">
            Oops... your browser doesn't support the HTML5 canvas element!
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 21, Col 29 Spaces: 4 UTF-8 CRLF HTML

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

drawPoints.html drawPoints.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawPoints.js > init > onclick

```
1   var gl;
2   var points, colors;
3
4   window.onload = function init() {
5       var canvas = document.getElementById("gl-canvas");
6
7       gl = WebGLUtils.setupWebGL(canvas);
8       if( !gl ) {
9           alert("WebGL isn't available!");
10      }
11
12      points = [];
13      colors = [];
14      var bMouseDown = false;
15
16      canvas.addEventListener("mousedown", function(event) {
17          bMouseDown = true;
18      });
19      canvas.addEventListener("mouseup", function(event) {
20          bMouseDown = false;
21      });
22      canvas.addEventListener("mousemove", function(event) {
23          if( bMouseDown ) {
24              var point = vec2(2 * event.clientX/canvas.width - 1,
25                  2 * (canvas.height - event.clientY) / canvas.height - 1);
26              points.push(point);
27              gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
28              gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
29
30              colors.push(currentColor);
31              gl.bindBuffer(gl.ARRAY_BUFFER, cbufferId);
32              gl.bufferData(gl.ARRAY_BUFFER, flatten(colors), gl.STATIC_DRAW);
33
34              render();
35          }
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 116, Col 11 Spaces: 4 UTF-8 CRLF JavaScript

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

drawPoints.html drawPoints.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawPoints.js > init > onclick

```
41
42      // Load shaders and initialize attribute buffers
43      var program = initShaders(gl, "vertex-shader", "fragment-shader");
44      gl.useProgram(program);
45
46      // Load the data into the GPU
47      var bufferId = gl.createBuffer();
48      gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
49      gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
50
51      // Associate our shader variables with our data buffer
52      var vPosition = gl.getAttribLocation(program, "vPosition");
53      gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
54      gl.enableVertexAttribArray(vPosition);
55
56      // Create a buffer object, initialize it, and associate it with
57      // the associated attribute variable in our vertex shader
58      var cbufferId = gl.createBuffer();
59      gl.bindBuffer(gl.ARRAY_BUFFER, cbufferId);
60      gl.bufferData(gl.ARRAY_BUFFER, flatten(colors), gl.STATIC_DRAW);
61
62      var vColor = gl.getAttribLocation(program, "vColor");
63      gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 0);
64      gl.enableVertexAttribArray(vColor);
65
66      var pointSize = gl.getUniformLocation(program, "pointSize");
67      gl.uniform1f(pointSize, 10.0);
68
69      document.getElementById("pointSize").onchange = function() {
70          var size = this.value;
71          gl.uniform1f(pointSize, size);
72
73          render();
74      }
75
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 116, Col 11 Spaces: 4 UTF-8 CRLF JavaScript

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

drawPoints.html drawPoints.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawPoints.js > init > onclick

```
75
76      //var fColor = gl.getUniformLocation(program, "fColor");
77      //gl.uniform4f(fColor, 1.0, 0.0, 0.0, 1.0);
78      var currentColor = vec4(1.0, 0.0, 0.0, 1.0);
79
80      document.getElementById("pointColor").onclick = function(event) {
81          switch(event.target.value) {
82              case "red":
83                  //gl.uniform4f(fColor, 1.0, 0.0, 0.0, 1.0);
84                  currentColor = vec4(1.0, 0.0, 0.0, 1.0);
85                  break;
86              case "green":
87                  //gl.uniform4f(fColor, 0.0, 1.0, 0.0, 1.0);
88                  currentColor = vec4(0.0, 1.0, 0.0, 1.0);
89                  break;
90              case "blue":
91                  //gl.uniform4f(fColor, 0.0, 0.0, 1.0, 1.0);
92                  currentColor = vec4(0.0, 0.0, 1.0, 1.0);
93                  break;
94              case "yellow":
95                  //gl.uniform4f(fColor, 1.0, 1.0, 0.0, 1.0);
96                  currentColor = vec4(1.0, 1.0, 0.0, 1.0);
97                  break;
98              case "cyan":
99                  //gl.uniform4f(fColor, 0.0, 1.0, 1.0, 1.0);
100                 currentColor = vec4(0.0, 1.0, 1.0, 1.0);
101                 break;
102             case "magenta":
103                 //gl.uniform4f(fColor, 1.0, 0.0, 1.0, 1.0);
104                 currentColor = vec4(1.0, 0.0, 1.0, 1.0);
105                 break;
106             case "gray":
107                 //gl.uniform4f(fColor, 0.5, 0.5, 0.5, 1.0);
108                 currentColor = vec4(0.5, 0.5, 0.5, 1.0);
109                 break;
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 116, Col 11 Spaces: 4 UTF-8 CRLF JavaScript

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

drawPoints.html | drawPoints.js

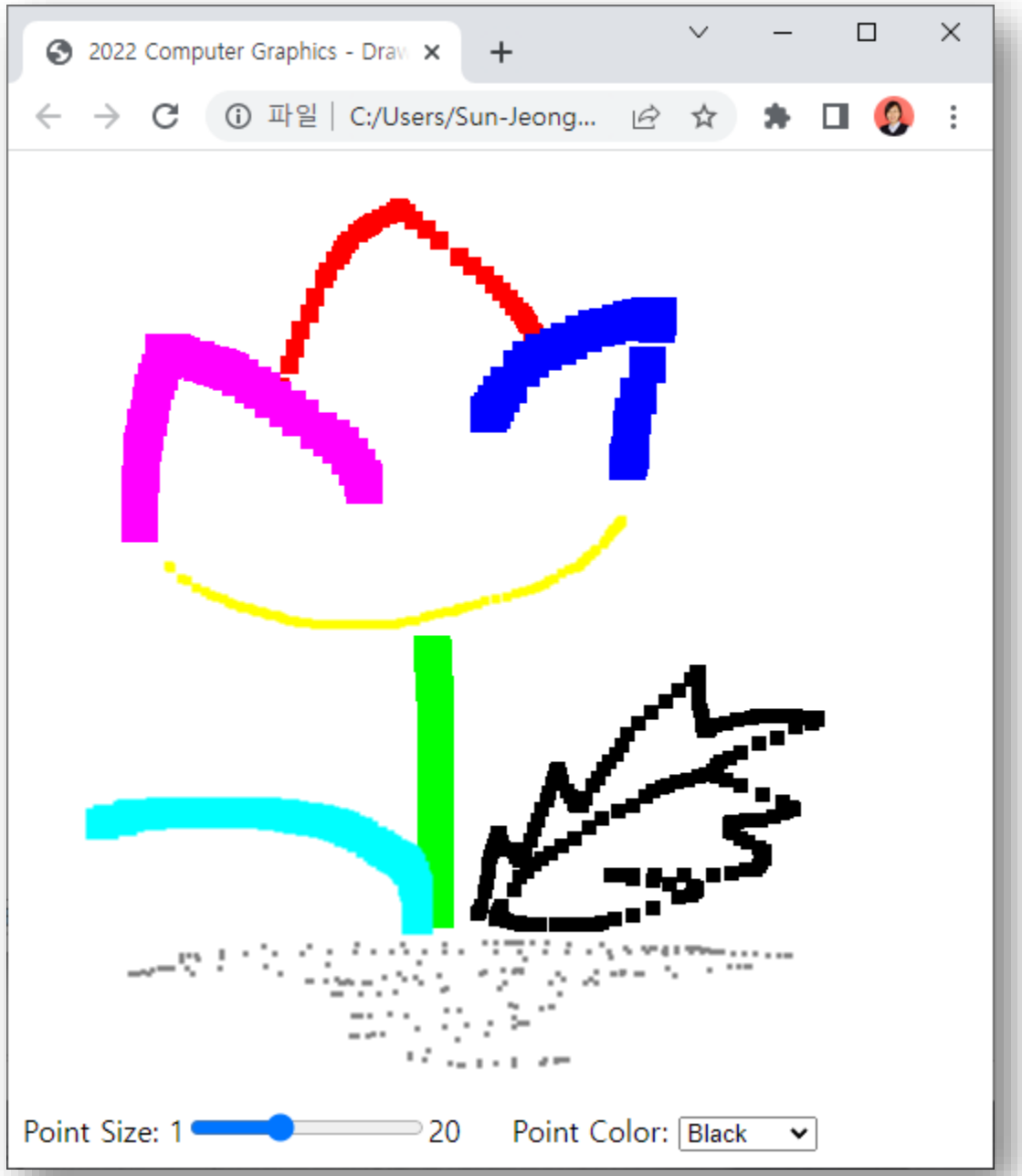
C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawPoints.js > init > onclick

```
98              case "cyan":
99                  //gl.uniform4f(fColor, 0.0, 1.0, 1.0, 1.0);
100                 currentColor = vec4(0.0, 1.0, 1.0, 1.0);
101                 break;
102             case "magenta":
103                 //gl.uniform4f(fColor, 1.0, 0.0, 1.0, 1.0);
104                 currentColor = vec4(1.0, 0.0, 1.0, 1.0);
105                 break;
106             case "gray":
107                 //gl.uniform4f(fColor, 0.5, 0.5, 0.5, 1.0);
108                 currentColor = vec4(0.5, 0.5, 0.5, 1.0);
109                 break;
110             case "black":
111                 //gl.uniform4f(fColor, 0.0, 0.0, 0.0, 1.0);
112                 currentColor = vec4(0.0, 0.0, 0.0, 1.0);
113                 break;
114         }
115
116         //render();
117     }
118
119     render();
120 };
121
122 function render() {
123     gl.clear(gl.COLOR_BUFFER_BIT);
124     gl.drawArrays(gl.POINTS, 0, points.length);
125 }
126
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 116, Col 11 Spaces: 4 UTF-8 CRLF JavaScript

연습 문제 (4)

- 점마다 크기를 다르게 설정하여 그리시오.



Linking Shader with Application

- Read shaders
- Compile shaders
- Create a program object
- Link everything together
- Link variables in application with variables in shaders
  - Vertex attributes
  - Uniform variables

Program Object

- Container for shaders
  - Can contain multiple shaders
  - Other GLSL functions

```
var program = gl.createProgram();

/* define shader objects here */

gl.attachShader( program, vertShdr );
gl.attachShader( program, fragShdr );
gl.linkProgram( program );
```

Reading a Shader

- Shaders are added to the program object and compiled
- Usual method of passing a shader is as a null-terminated string using the function `glShaderSource( shdr, text );`
- If the shader is in HTML file, we can get it into application by `getElementById` method
- If the shader is in a file, we can write a reader to convert the file to a string

Adding a Vertex Shader

```
var vertShdr;
var vertElem = document.getElementById(vertexShaderId);

vertShdr = gl.createShader( gl.VERTEX_SHADER );

gl.shaderSource( vertShdr, vertElem.text );
gl.compileShader( vertShdr );

// after program object created
gl.attachShader( program, vertShdr );
```

Shader Reader

- Following code may be a security issue with some browsers if you try to run it locally
  - Cross origin request

```
function getShader( gl, shaderName, type ) {
    var shader = gl.createShader( type );
    shaderScript = loadFileAJAX( shaderName );
    if( !shaderScript ) {
        alert( "Could not find shader source:"
               + shaderName );
    }
}
```

Precision Declaration

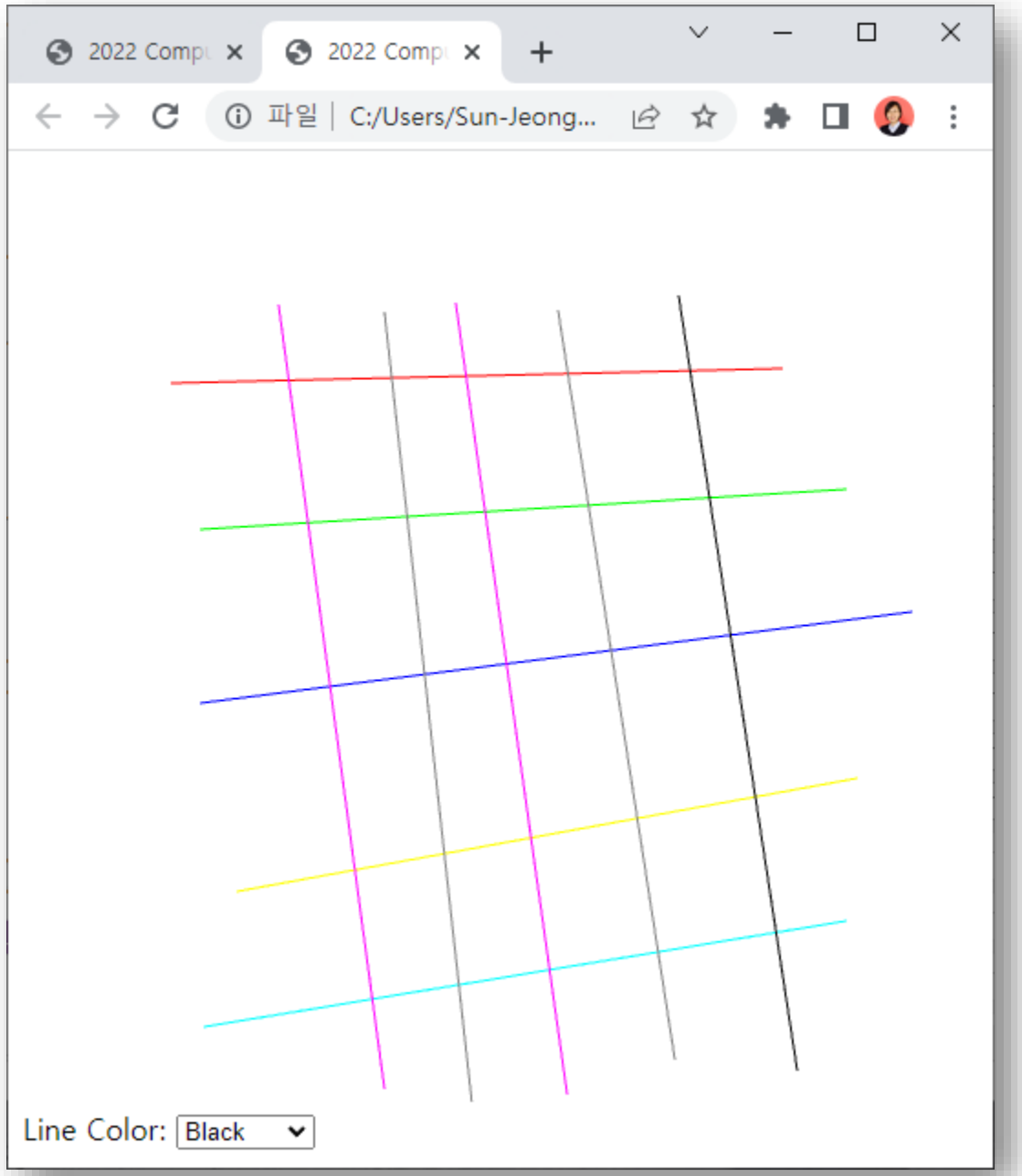
- In GLSL for WebGL we must specify desired precision in fragment shaders
  - Artifact inherited from OpenGL ES
  - ES must run on very simple embedded devices that may not support 32-bit floating point
  - All implementations must support mediump
  - No default for float in fragment shader
- Can use preprocessor directives (#ifdef) to check if highp supported and, if not, default to mediump

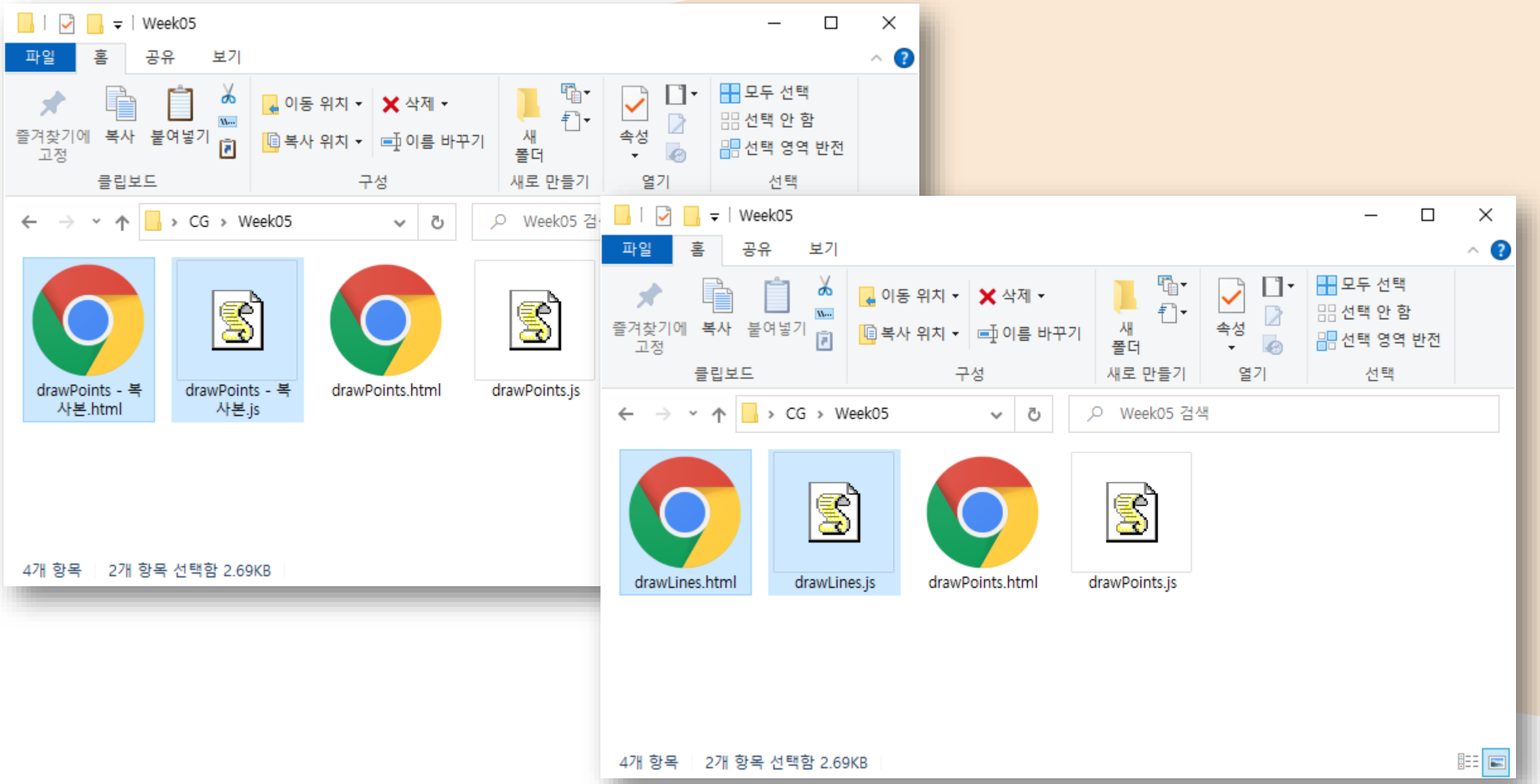
Pass Through Fragment Shader

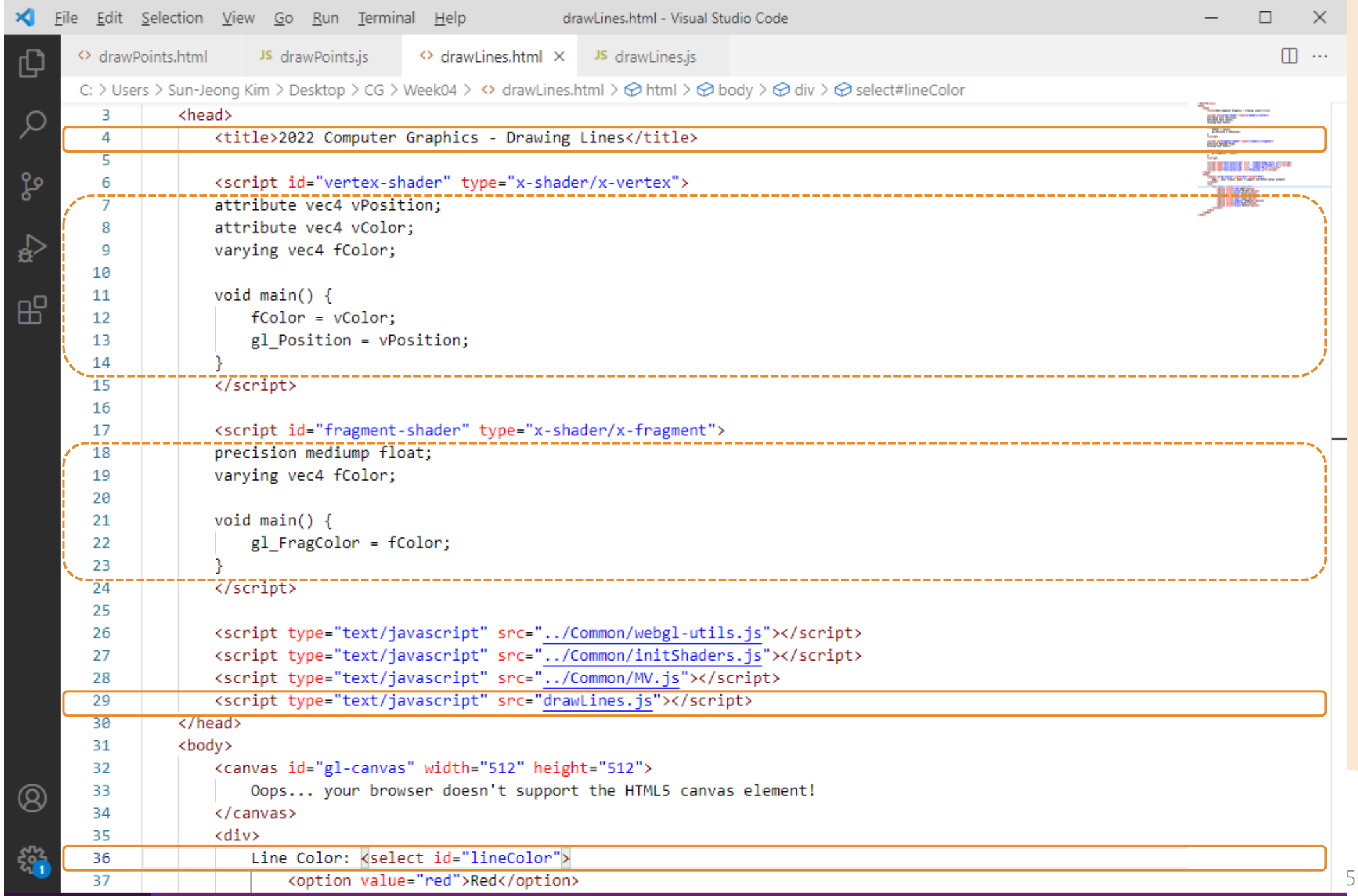
```
#ifdef GL_FRAGMENT_SHADER_PRECISION_HIGH
precision highp float;
#else
precision mediump float;
#endif

varying vec4 fColor;
void main(void) {
    gl_FragColor = fColor;
}
```

연습 문제 (5)

- Drag하여 선분들을 그리시오.




Week05
파일
홈
공유
보기
즐겨찾기에 고정
복사
붙여넣기
이동 위치
복사 위치
삭제
이름 바꾸기
새 폴더
속성
모두 선택
선택 안 함
선택 영역 반전
클립보드
구성
새로 만들기
열기
선택
CG › Week05
Week05 검색
drawPoints - 복사본.html
drawPoints - 복사본.js
drawPoints.html
drawPoints.js
4개 항목
2개 항목 선택함 2.69KB
Week05
파일
홈
공유
보기
즐겨찾기에 고정
복사
붙여넣기
이동 위치
복사 위치
삭제
이름 바꾸기
새 폴더
속성
모두 선택
선택 안 함
선택 영역 반전
클립보드
구성
새로 만들기
열기
선택
CG › Week05
Week05 검색
drawLines.html
drawLines.js
drawPoints.html
drawPoints.js
4개 항목
2개 항목 선택함 2.69KB


File Edit Selection View Go Run Terminal Help
drawLines.html - Visual Studio Code
drawPoints.html
drawPoints.js
drawLines.html
drawLines.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawLines.html > html > body > div > select#lineColor
3 <head>
4 <title>2022 Computer Graphics - Drawing Lines</title>
5
6 <script id="vertex-shader" type="x-shader/x-vertex">
7 attribute vec4 vPosition;
8 attribute vec4 vColor;
9 varying vec4 fColor;
10
11 void main() {
12 fColor = vColor;
13 gl_Position = vPosition;
14 }
15 </script>
16
17 <script id="fragment-shader" type="x-shader/x-fragment">
18 precision mediump float;
19 varying vec4 fColor;
20
21 void main() {
22 gl_FragColor = fColor;
23 }
24 </script>
25
26 <script type="text/javascript" src="../Common/webgl-utils.js"></script>
27 <script type="text/javascript" src="../Common/initShaders.js"></script>
28 <script type="text/javascript" src="../Common/MV.js"></script>
29 <script type="text/javascript" src="drawLines.js"></script>
30 </head>
31 <body>
32 <canvas id="gl-canvas" width="512" height="512">
33 Oops... your browser doesn't support the HTML5 canvas element!
34 </canvas>
35
36 Line Color: <select id="lineColor">
37 <option value="red">Red</option>
Restricted Mode
0 0
Ln 36, Col 41
Spaces: 4
UTF-8
CRLF
HTML

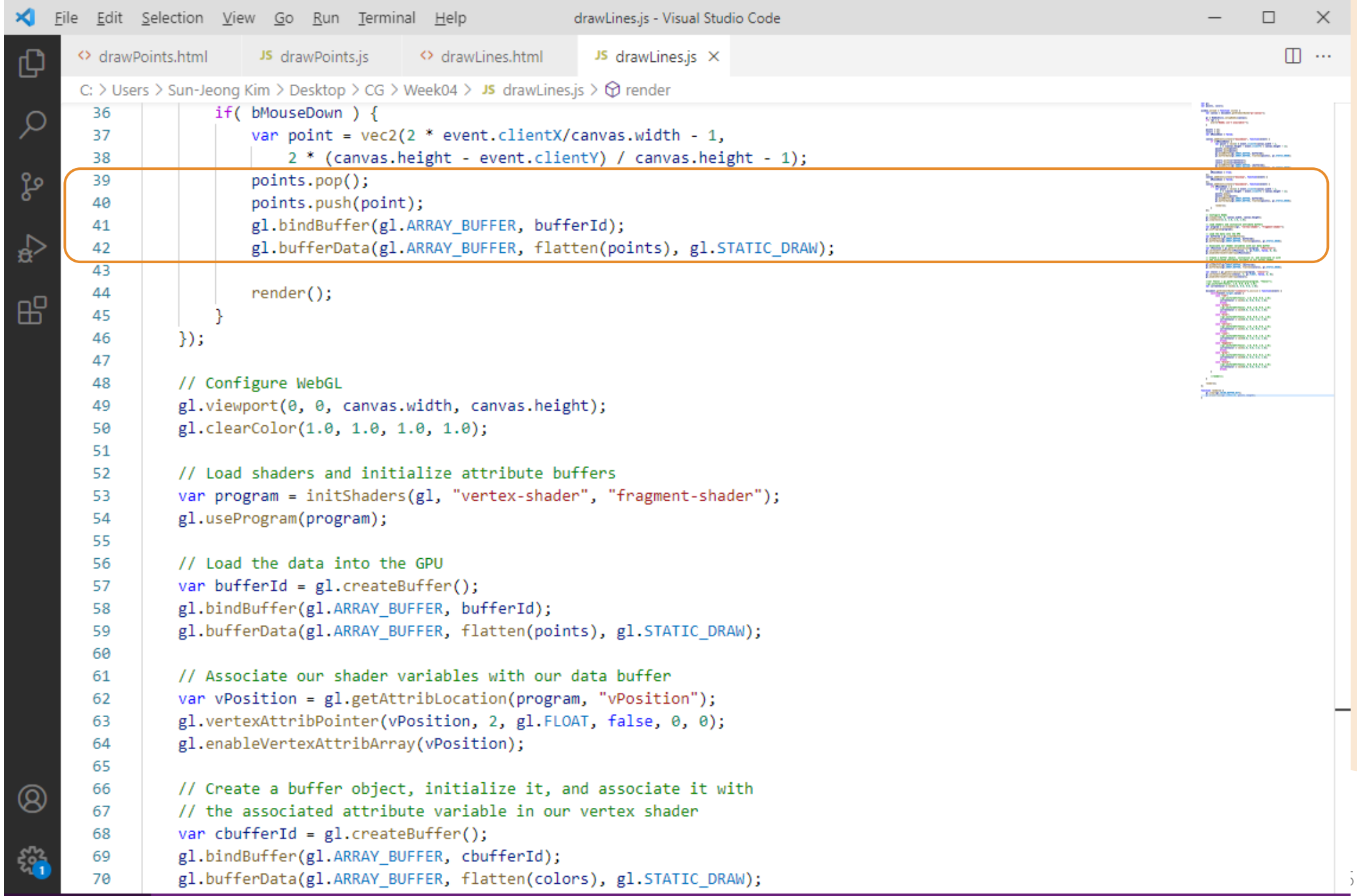
File Edit Selection View Go Run Terminal Help drawLines.js - Visual Studio Code

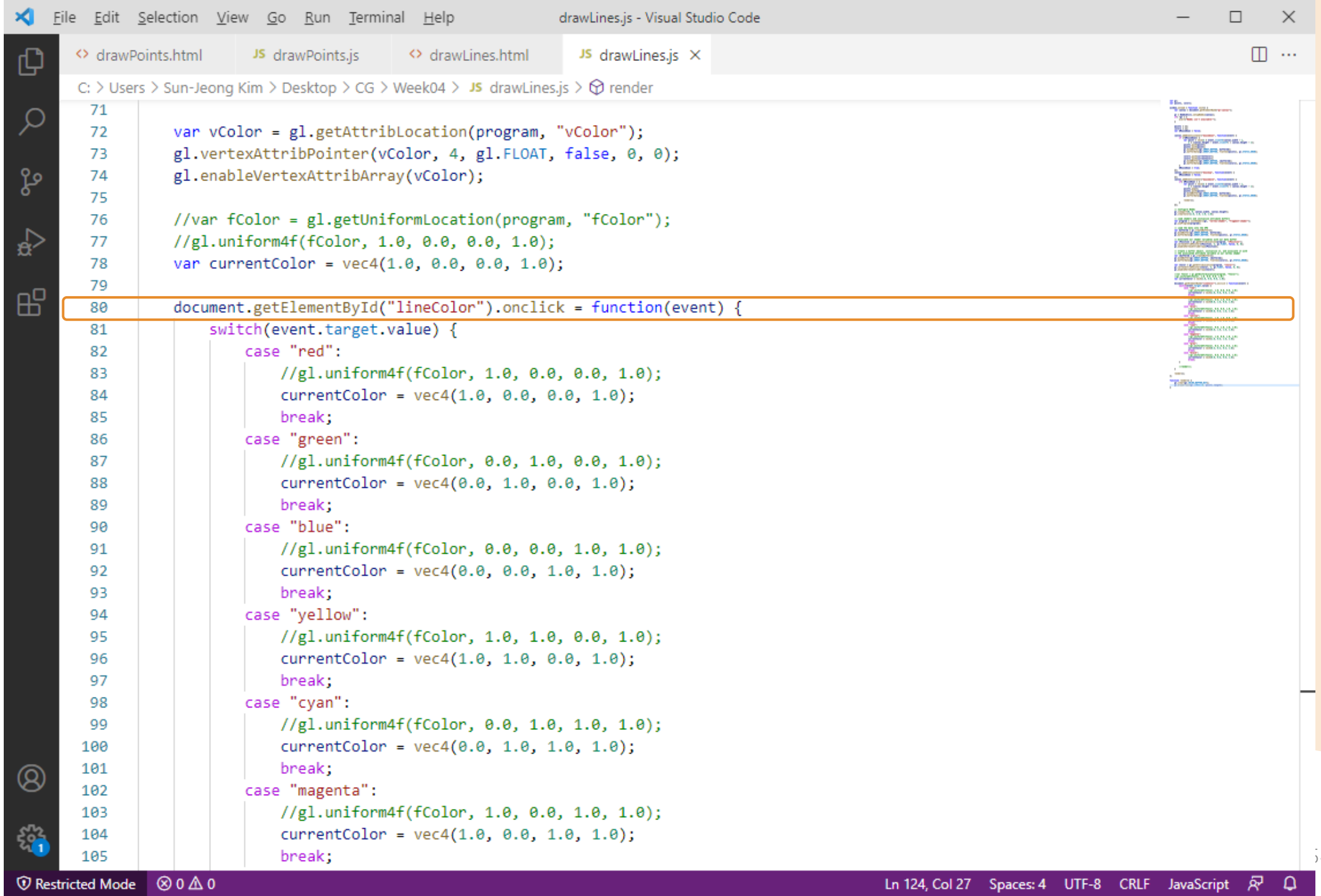
drawPoints.html | drawPoints.js | drawLines.html | drawLines.js

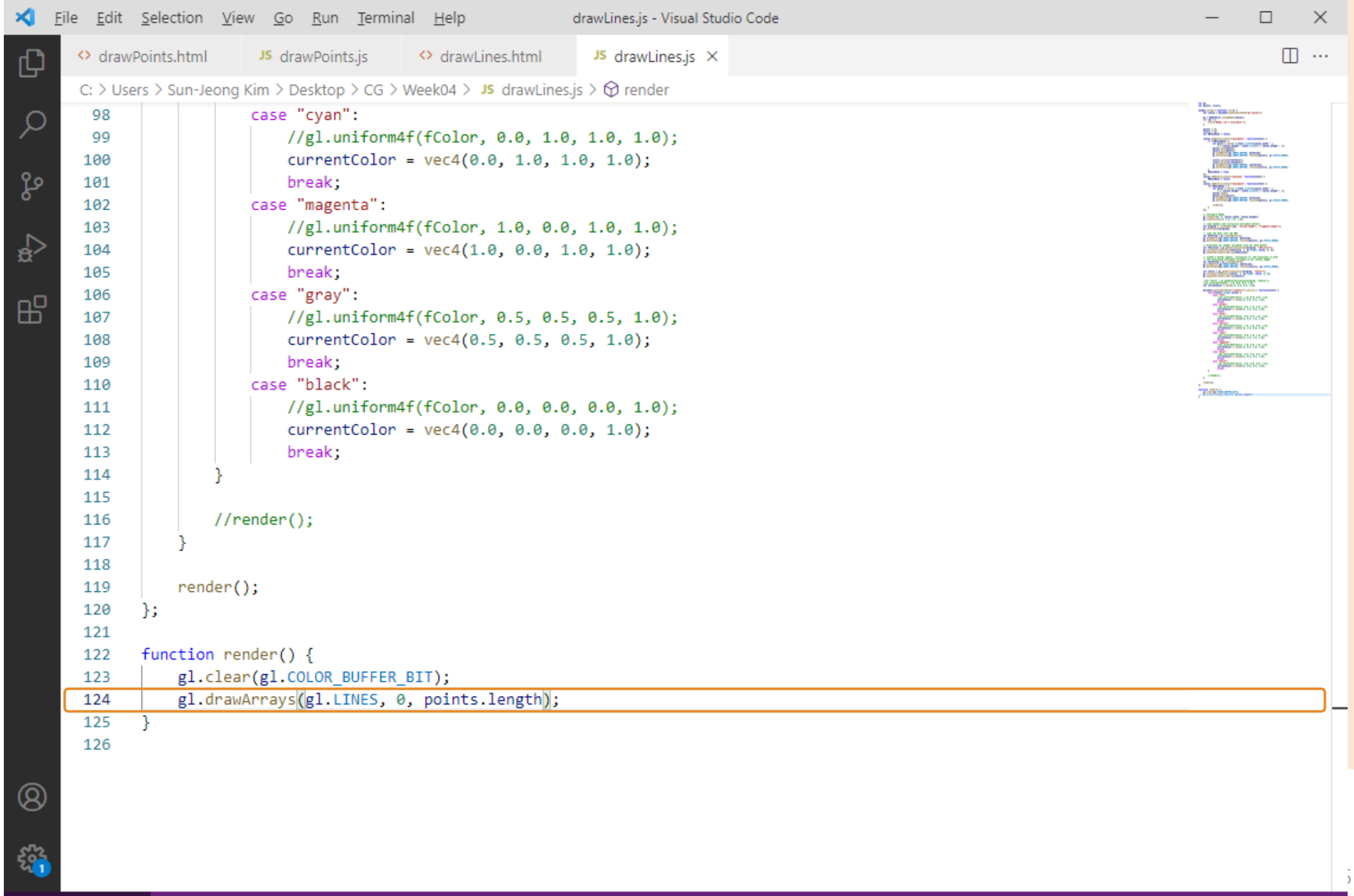
C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawLines.js > render

```
1   var gl;
2   var points, colors;
3
4   window.onload = function init() {
5       var canvas = document.getElementById("gl-canvas");
6
7       gl = WebGLUtils.setupWebGL(canvas);
8       if( !gl ) {
9           alert("WebGL isn't available!");
10      }
11
12      points = [];
13      colors = [];
14      var bMouseDown = false;
15
16      canvas.addEventListener("mousedown", function(event) {
17          if (!bMouseDown) {
18              var point = vec2(2 * event.clientX/canvas.width - 1,
19                  2 * (canvas.height - event.clientY) / canvas.height - 1);
20              points.push(point);
21              points.push(point);
22              gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
23              gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
24
25              colors.push(currentColor);
26              colors.push(currentColor);
27              gl.bindBuffer(gl.ARRAY_BUFFER, cbufferId);
28              gl.bufferData(gl.ARRAY_BUFFER, flatten(colors), gl.STATIC_DRAW);
29          }
30          bMouseDown = true;
31      });
32      canvas.addEventListener("mouseup", function(event) {
33          bMouseDown = false;
34      });
35      canvas.addEventListener("mousemove", function(event) {
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 124, Col 27 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
drawLines.js - Visual Studio Code
drawPoints.html
drawPoints.js
drawLines.html
drawLines.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawLines.js > render
36 if(bMouseDown) {
37 var point = vec2(2 * event.clientX/canvas.width - 1,
38 2 * (canvas.height - event.clientY) / canvas.height - 1);
39 points.pop();
40 points.push(point);
41 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
42 gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
43
44 render();
45 }
46 });
47
48 // Configure WebGL
49 gl.viewport(0, 0, canvas.width, canvas.height);
50 gl.clearColor(1.0, 1.0, 1.0, 1.0);
51
52 // Load shaders and initialize attribute buffers
53 var program = initShaders(gl, "vertex-shader", "fragment-shader");
54 gl.useProgram(program);
55
56 // Load the data into the GPU
57 var bufferId = gl.createBuffer();
58 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
59 gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
60
61 // Associate our shader variables with our data buffer
62 var vPosition = gl.getAttribLocation(program, "vPosition");
63 gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
64 gl.enableVertexAttribArray(vPosition);
65
66 // Create a buffer object, initialize it, and associate it with
67 // the associated attribute variable in our vertex shader
68 var cbufferId = gl.createBuffer();
69 gl.bindBuffer(gl.ARRAY_BUFFER, cbufferId);
70 gl.bufferData(gl.ARRAY_BUFFER, flatten(colors), gl.STATIC_DRAW);
Restricted Mode
Ln 124, Col 27 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
drawLines.js - Visual Studio Code
drawPoints.html
drawPoints.js
drawLines.html
drawLines.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawLines.js > render
71
72 var vColor = gl.getAttribLocation(program, "vColor");
73 gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 0);
74 gl.enableVertexAttribArray(vColor);
75
76 //var fColor = gl.getUniformLocation(program, "fColor");
77 //gl.uniform4f(fColor, 1.0, 0.0, 0.0, 1.0);
78 var currentColor = vec4(1.0, 0.0, 0.0, 1.0);
79
80 document.getElementById("lineColor").onclick = function(event) {
81 switch(event.target.value) {
82 case "red":
83 //gl.uniform4f(fColor, 1.0, 0.0, 0.0, 1.0);
84 currentColor = vec4(1.0, 0.0, 0.0, 1.0);
85 break;
86 case "green":
87 //gl.uniform4f(fColor, 0.0, 1.0, 0.0, 1.0);
88 currentColor = vec4(0.0, 1.0, 0.0, 1.0);
89 break;
90 case "blue":
91 //gl.uniform4f(fColor, 0.0, 0.0, 1.0, 1.0);
92 currentColor = vec4(0.0, 0.0, 1.0, 1.0);
93 break;
94 case "yellow":
95 //gl.uniform4f(fColor, 1.0, 1.0, 0.0, 1.0);
96 currentColor = vec4(1.0, 1.0, 0.0, 1.0);
97 break;
98 case "cyan":
99 //gl.uniform4f(fColor, 0.0, 1.0, 1.0, 1.0);
100 currentColor = vec4(0.0, 1.0, 1.0, 1.0);
101 break;
102 case "magenta":
103 //gl.uniform4f(fColor, 1.0, 0.0, 1.0, 1.0);
104 currentColor = vec4(1.0, 0.0, 1.0, 1.0);
105 break;
Restricted Mode
0 0
Ln 124, Col 27
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
drawLines.js - Visual Studio Code
drawPoints.html
drawPoints.js
drawLines.html
drawLines.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > drawLines.js > render
98 case "cyan":
99 //gl.uniform4f(fColor, 0.0, 1.0, 1.0, 1.0);
100 currentColor = vec4(0.0, 1.0, 1.0, 1.0);
101 break;
102 case "magenta":
103 //gl.uniform4f(fColor, 1.0, 0.0, 1.0, 1.0);
104 currentColor = vec4(1.0, 0.0, 1.0, 1.0);
105 break;
106 case "gray":
107 //gl.uniform4f(fColor, 0.5, 0.5, 0.5, 1.0);
108 currentColor = vec4(0.5, 0.5, 0.5, 1.0);
109 break;
110 case "black":
111 //gl.uniform4f(fColor, 0.0, 0.0, 0.0, 1.0);
112 currentColor = vec4(0.0, 0.0, 0.0, 1.0);
113 break;
114 }
115
116 //render();
117 }
118
119 render();
120 };
121
122 function render() {
123 gl.clear(gl.COLOR_BUFFER_BIT);
124 gl.drawArrays(gl.LINES, 0, points.length);
125 }
126
Restricted Mode
0 0
Ln 124, Col 27
Spaces: 4
UTF-8
CRLF
JavaScript

연습 문제 (6)

- 세 점을 클릭하여 삼각형들을 그리시오.

- “drawTriangles.html”과 “drawTriangles.js”를 제출하시오.
- 결과 스크린샷 이미지도 함께 제출하시오.