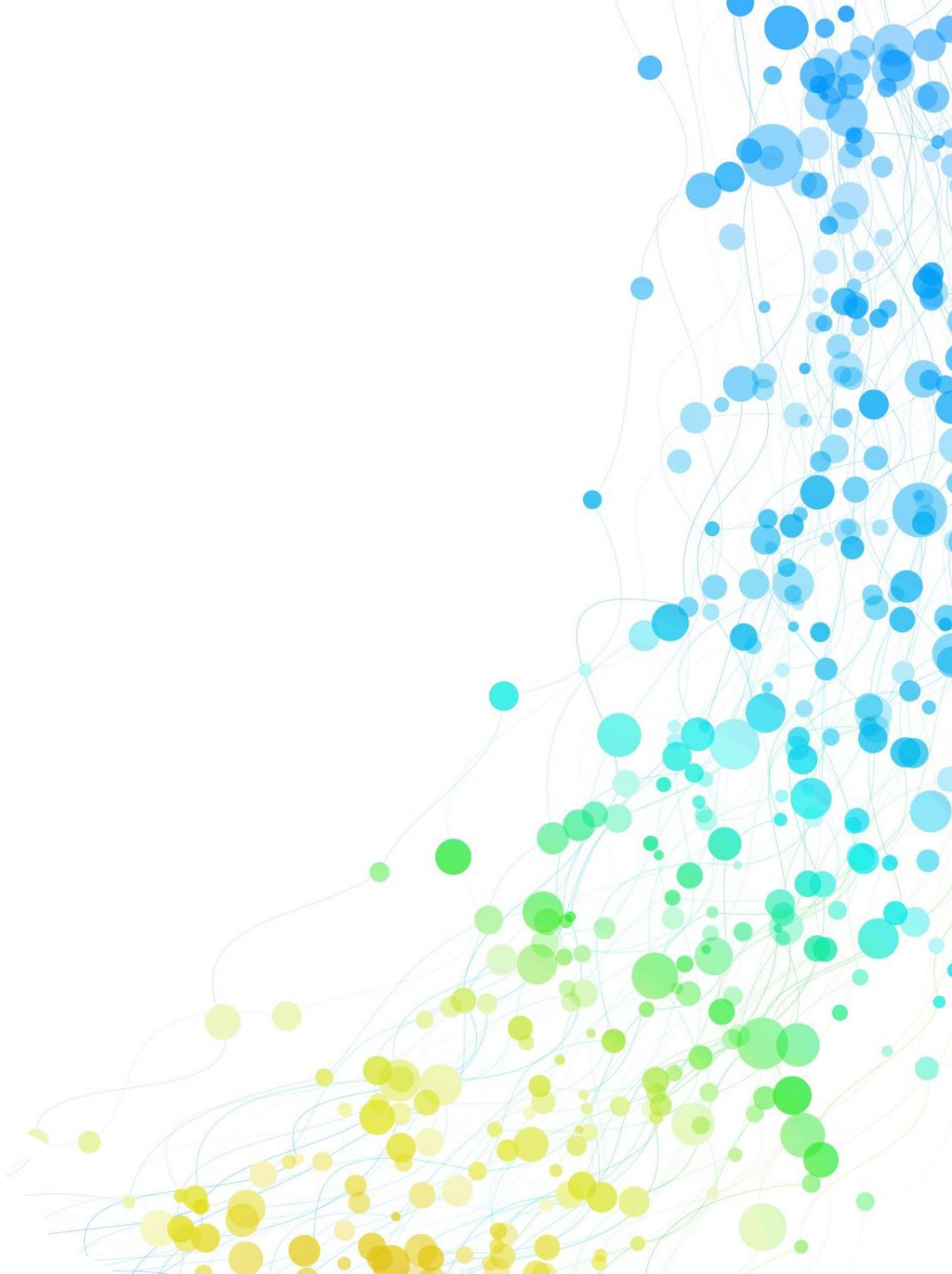
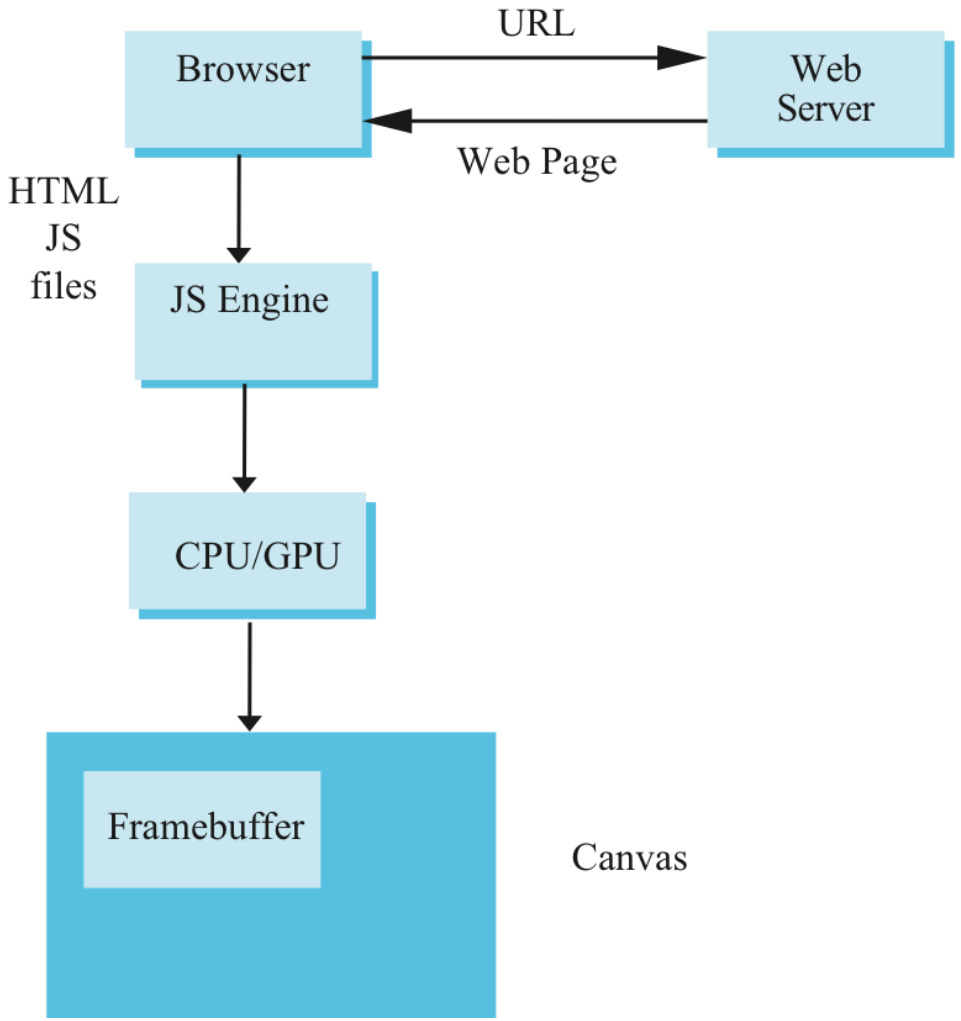
Interaction and Animation

4TH WEEK, 2021



Execution in Browser



Execution in Browser

- Start with HTML file
  - Describe the page
  - May contain the shaders
  - Loads files
- Files are loaded asynchronously and JS code executed
- Then what?
- Browser is in an event loop and waits for an event

Event Types

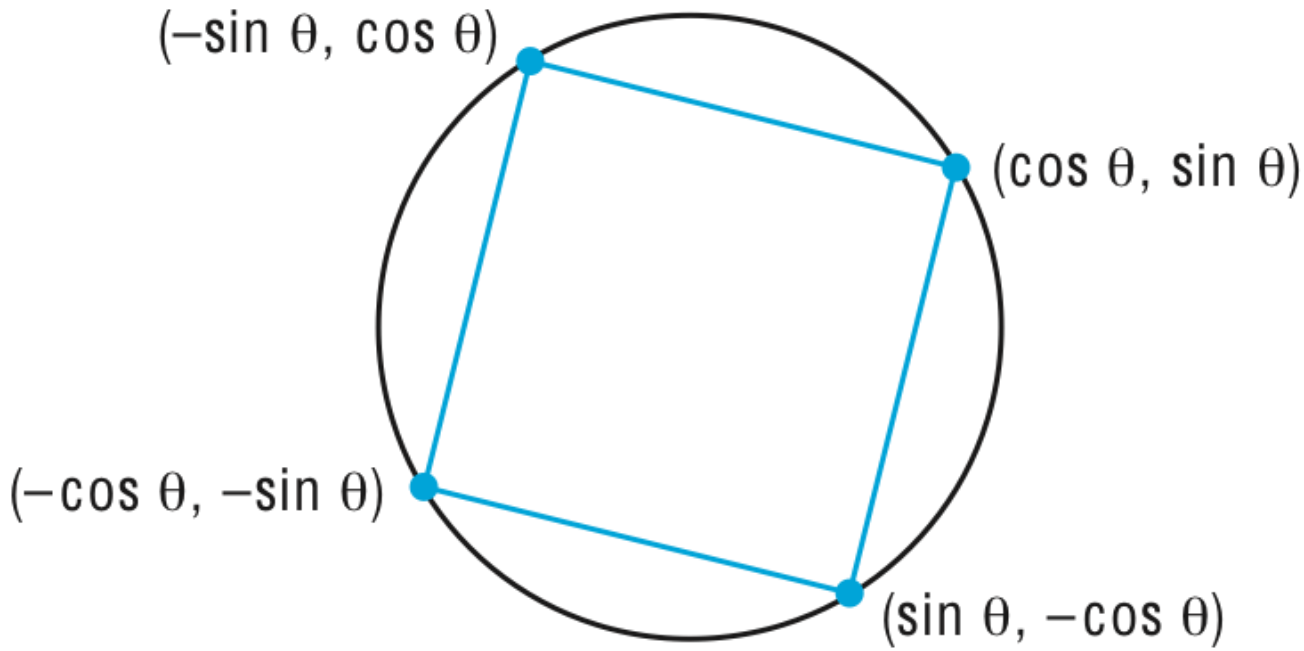
- Window: resize, expose, iconify
- Mouse: click one or more buttons
- Motion: move mouse
- Keyboard: press or release a key
- Idle: nonevent
  - Define what should be done if no other event is in queue

Callbacks

- Programming interface for event-driven input uses callback functions or event listeners
  - Define a callback for each event the graphics system recognizes
  - Browsers enters an event loop and responds to those events for which it has callbacks registered
  - The callback function is executed when the event occurs

Rotating Square

- Consider the four points



- Animate display by rendering with different value of θ

Simple but Slow Method

```
for(var theta = 0.0; theta<thetaMax; theta += dtheta) {

        vertices[0] = vec2(Math.sin(theta), Math.cos.(theta));
        vertices[1] = vec2(Math.sin(theta), -Math.cos.(theta));
        vertices[2] = vec2(-Math.sin(theta), -Math.cos.(theta));
        vertices[3] = vec2(-Math.sin(theta), Math.cos.(theta));

        gl.bufferSubData(.....................

        render();
}
```

Better Way

- Send original vertices to vertex shader
- Send θ to shader as a uniform variable
- Compute vertices in vertex shader
- Render recursively

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

rotateSquare.html

C: > Users > sunje > Desktop > CG > rotateSquare.html > html > head > script

```
<!DOCTYPE html>
<html>
    <head>
        <title>학번 이름</title>
        <script id="vertex-shader" type="x-shader/x-vertex">
        attribute vec4 vPosition;
        uniform float theta;

        void main() {
            float s = sin(theta);
            float c = cos(theta);
            gl_Position.x = c * vPosition.x - s * vPosition.y;
            gl_Position.y = s * vPosition.x + c * vPosition.y;
            gl_Position.z = 0.0;
            gl_Position.w = 1.0;
        }
        </script>

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

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

0 0 Ln 30, Col 57 Spaces: 4 UTF-8 CRLF HTML

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

rotateSquare.html

C: > Users > sunje > Desktop > CG > rotateSquare.html > html > head > script

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

void main() {
    float s = sin(theta);
    float c = cos(theta);
    gl_Position.x = c * vPosition.x - s * vPosition.y;
    gl_Position.y = s * vPosition.x + c * vPosition.y;
    gl_Position.z = 0.0;
    gl_Position.w = 1.0;
}
</script>

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

void main() {
    gl_FragColor = vec4(0.0, 1.0, 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="rotateSquare.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>
```

0 0 Ln 30, Col 57 Spaces: 4 UTF-8 CRLF HTML

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

rotateSquare.html | rotateSquare.js

C: > Users > sunje > Desktop > CG > rotateSquare.js > render

```
var gl;
var theta = 0;
var thetaLoc;

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

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

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

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

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

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

    // Associate our shader variables with our data buffer
```

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

rotateSquare.html | rotateSquare.js

C: > Users > sunje > Desktop > CG > rotateSquare.js > render

```
21
22      // Configure WebGL
23      gl.viewport(0, 0, canvas.width, canvas.height);
24      gl.clearColor(1.0, 1.0, 1.0, 1.0);
25
26      // Load shaders and initialize attribute buffers
27      var program = initShaders(gl, "vertex-shader", "fragment-shader");
28      gl.useProgram(program);
29
30      // Load the data into the GPU
31      var bufferId = gl.createBuffer();
32      gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
33      gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);
34
35      // Associate our shader variables with our data buffer
36      var vPosition = gl.getAttribLocation(program, "vPosition");
37      gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
38      gl.enableVertexAttribArray(vPosition);
39
40      thetaLoc = gl.getUniformLocation(program, "theta");
41      //gl.uniform1f(thetaLoc, theta);
42
43      render();
44  };
45
46  function render() {
47      gl.clear(gl.COLOR_BUFFER_BIT);
48
49      theta += 0.1;
50      gl.uniform1f(thetaLoc, theta);
51
52      gl.drawArrays(gl.TRIANGLE_STRIP, 0, 4);
53  }
54
```

⊗ 0 ⚠ 0 Ln 52, Col 36 Spaces: 4 UTF-8 CRLF JavaScript

Double Buffering

- Although we are rendering the square, it always into a buffer that is not displayed
- Browser uses double buffering
  - Always display front buffer
  - Rendering into back buffer
  - Need a buffer swap
- Prevents display of a partial rendering

Triggering a Buffer Swap

- Browsers refresh the display at ~ 60 Hz
  - Redisplay of front buffer
  - Not a buffer swap
- Trigger a buffer swap through an event
- Two options for rotating square
  - Interval timer
  - requestAnimFrame

Interval Timer

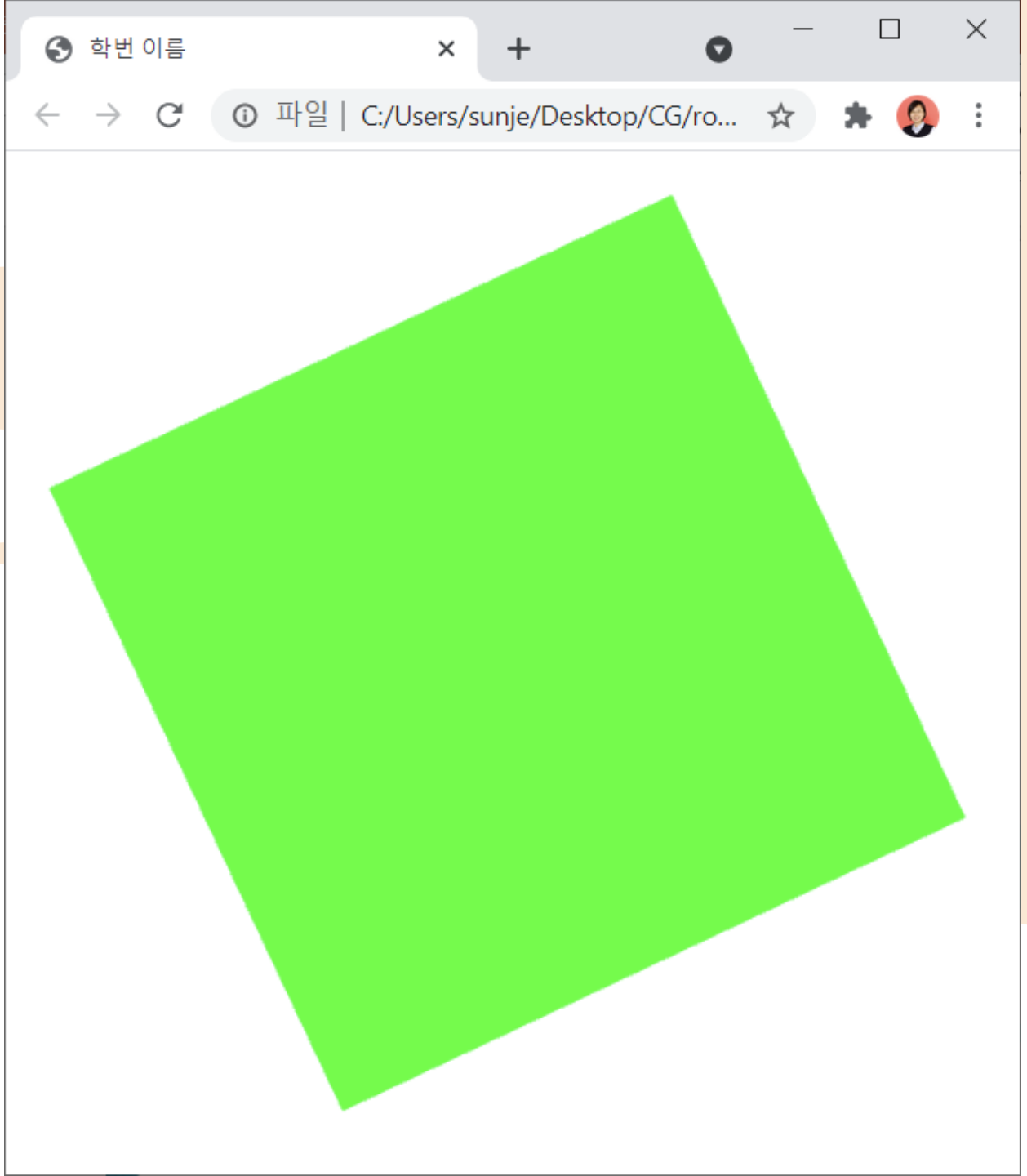
- Executes a function after a specified number of milliseconds
  - Also generates a buffer swap

```
setInterval(render, interval);
```

- Note an interval of 0 generates buffer swaps as fast as possible

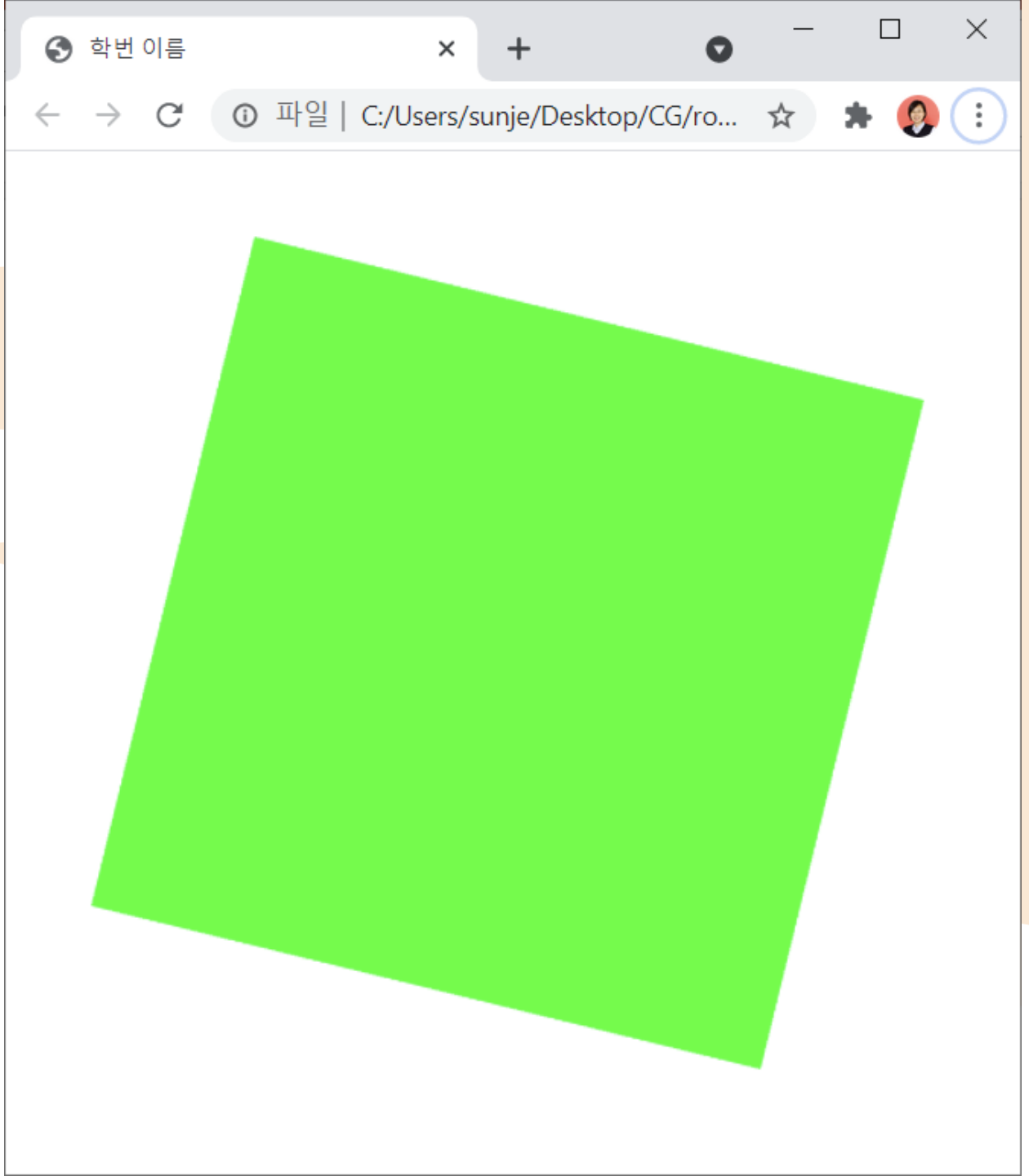
requestAnimFrame

```
46  function render() {
47      gl.clear(gl.COLOR_BUFFER_BIT);
48
49      theta += 0.1;
50      gl.uniform1f(thetaLoc, theta);
51
52      gl.drawArrays(gl.TRIANGLE_STRIP, 0, 4);
53
54      window.requestAnimationFrame(render);
55  }
```


학번 이름
파일 | C:/Users/sunje/Desktop/CG/ro...

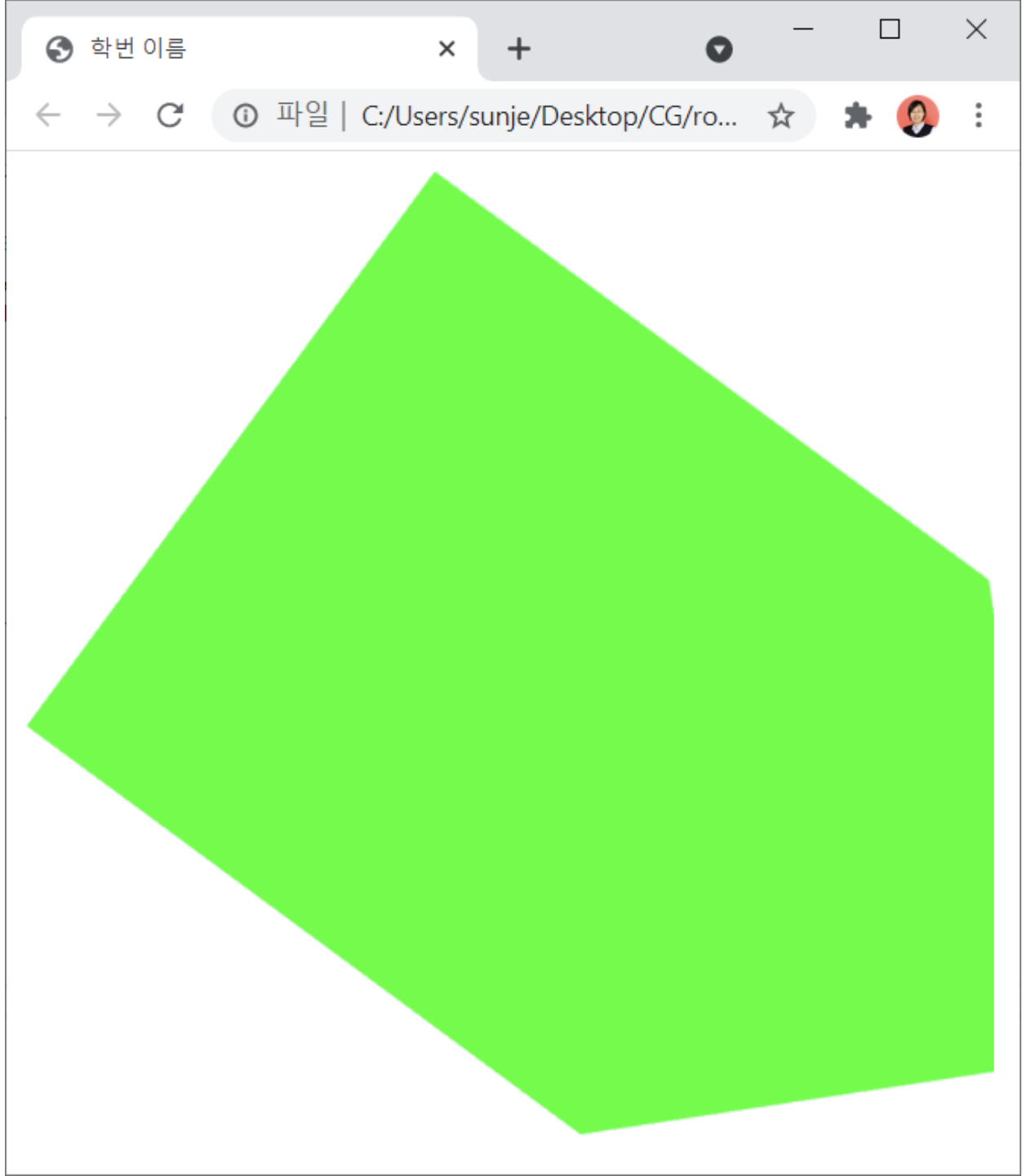
Add an Interval

```
46  function render() {
47      setTimeout(function() {
48          gl.clear(gl.COLOR_BUFFER_BIT);
49
50          theta += 0.1;
51          gl.uniform1f(thetaLoc, theta);
52
53          gl.drawArrays(gl.TRIANGLE_STRIP, 0, 4);
54
55          window.requestAnimationFrame(render);
56      }, 100);
57  }
```


학번 이름
파일 | C:/Users/sunje/Desktop/CG/ro...

연습 문제 (1)

- 오각형을 회전 시키시오.



Adding a Button

- Let's add a button to control the rotation direction for our rotating square
- In the render function we can use a **`var direction`** which is true or false to add or subtract a constant to the angle

```
var direction = true; // global initialization

// in render()

if(direction) theta += 0.1;
else theta -= 0.1;
```

The Button

- In the HTML file

```
<button id="DirectionButton">Change Rotation Direction</button>
```

  - Uses HTML button tag
  - id gives an identifier we can use in JS file
  - Text “Change Rotation Direction” displayed in button
- Clicking on button generates a click event
- Note we are using default style and could use CSS or jQuery to get a prettier button

Button Event Listener

- We still need to define the listener
  - No listener and the event occurs but is ignored
- Two forms for event listener in JS file

```
var myButton = document.getElementById("DirectionButton");

myButton.addEventListener("click", function() {
    direction = !direction;
});
```

```
document.getElementById("DirectionButton").onclick = function() {
    direction = !direction;
};
```

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

rotateSquare.html rotateSquare.js

C: > Users > sunje > Desktop > CG > rotateSquare.html > html > body > button#DirectionButton

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

void main() {
    float s = sin(theta);
    float c = cos(theta);
    gl_Position.x = c * vPosition.x - s * vPosition.y;
    gl_Position.y = s * vPosition.x + c * vPosition.y;
    gl_Position.z = 0.0;
    gl_Position.w = 1.0;
}
</script>

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

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

⊗ 0 ⚠ 0 Ln 33, Col 63 Spaces: 4 UTF-8 CRLF HTML

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

rotateSquare.html | rotateSquare.js

C: > Users > sunje > Desktop > CG > rotateSquare.js > ...

```
var gl;
var theta = 0;
var thetaLoc;
var direction = true;

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

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

    // Initialize event handlers
    document.getElementById("DirectionButton").onclick = function() {
        direction = !direction;
    }

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

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

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

```

0 0 Ln 64, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

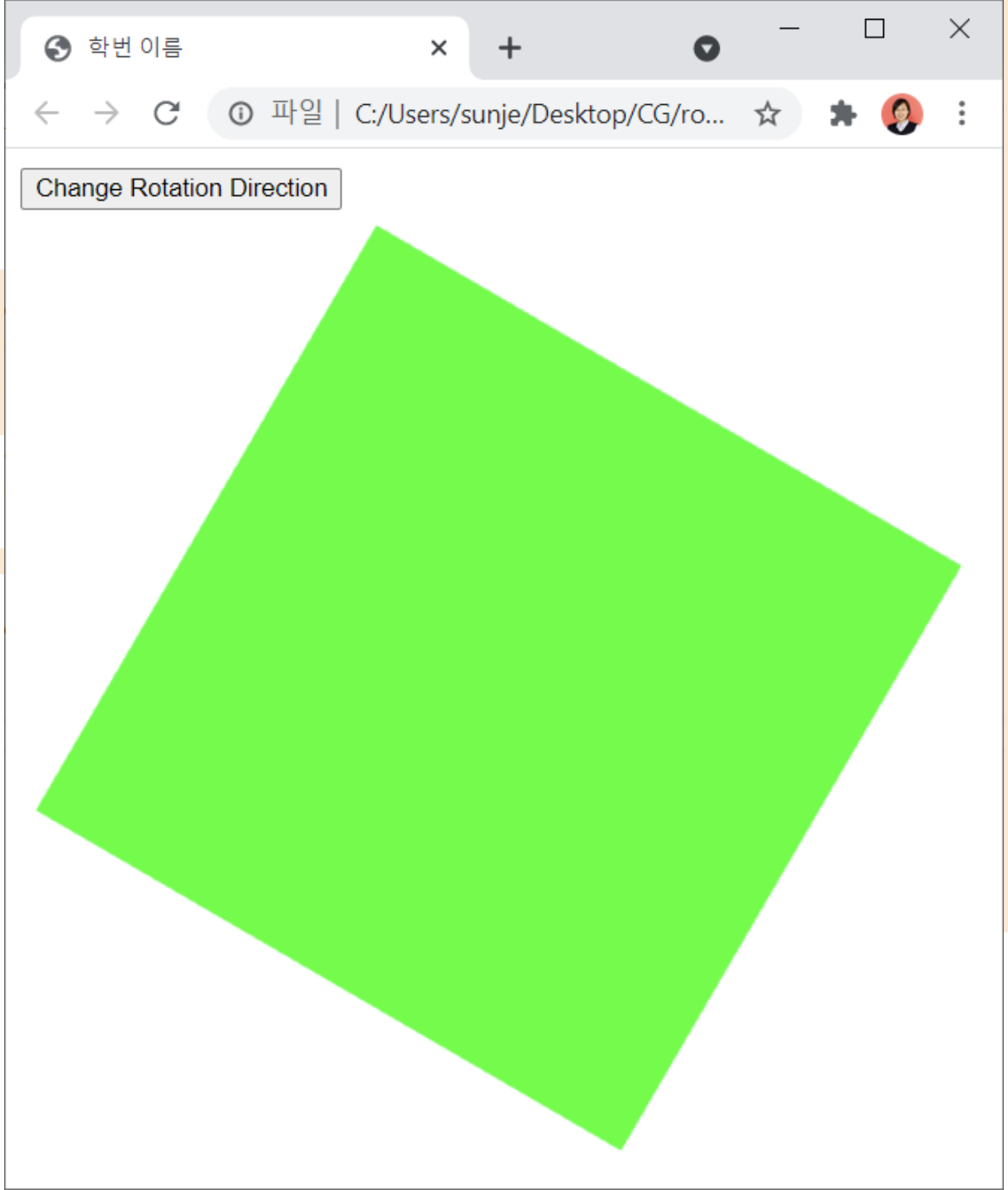
File Edit Selection View Go Run Terminal Help rotateSquare.js - Visual Studio Code

rotateSquare.html | rotateSquare.js

C: > Users > sunje > Desktop > CG > rotateSquare.js > render > setTimeout() callback

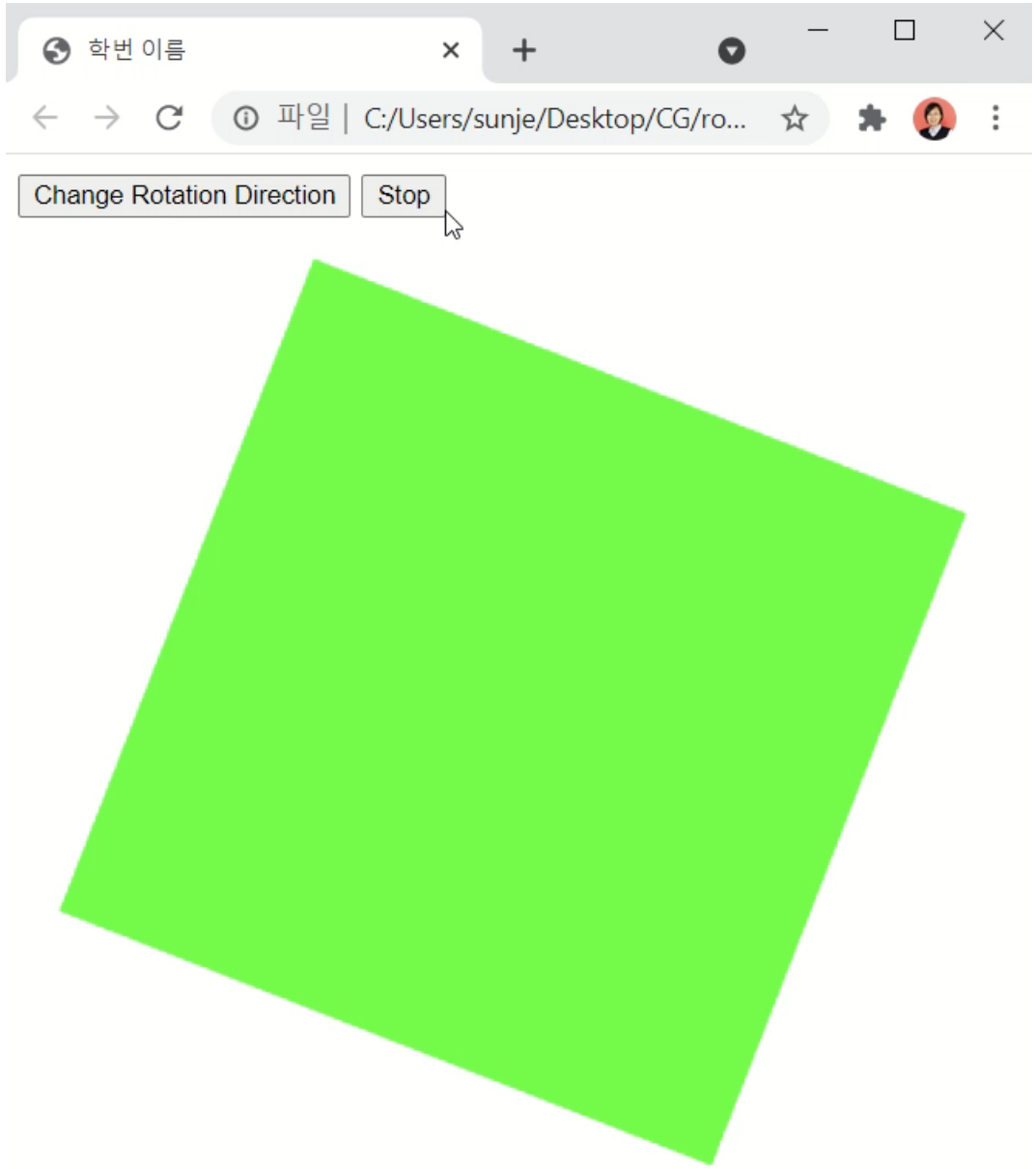
```
31
32      // Load shaders and initialize attribute buffers
33      var program = initShaders(gl, "vertex-shader", "fragment-shader");
34      gl.useProgram(program);
35
36      // Load the data into the GPU
37      var bufferId = gl.createBuffer();
38      gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
39      gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);
40
41      // Associate our shader variables with our data buffer
42      var vPosition = gl.getAttribLocation(program, "vPosition");
43      gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
44      gl.enableVertexAttribArray(vPosition);
45
46      thetaLoc = gl.getUniformLocation(program, "theta");
47      //gl.uniform1f(thetaLoc, theta);
48
49      render();
50  };
51
52  function render() {
53      setTimeout(function() {
54          gl.clear(gl.COLOR_BUFFER_BIT);
55
56          theta += (direction ? 0.1 : -0.1);
57          gl.uniform1f(thetaLoc, theta);
58
59          gl.drawArrays(gl.TRIANGLE_STRIP, 0, 4);
60
61          window.requestAnimationFrame(render);
62      }, 100);
63  }
64
```

⊗ 0 ⚠ 0 Ln 56, Col 41 Spaces: 4 UTF-8 CRLF JavaScript


학번 이름
파일 | C:/Users/sunje/Desktop/CG/ro...
Change Rotation Direction

연습 문제 (2)

- 회전을 멈추거나 시작하는 Toggle 버튼을 만드시오.



Menus

- Use the HTML select element
- Each entry in the menu is an option element with an integer value returned by click event

```
32      <body>
33          <button id="DirectionButton">Change Rotation Direction</button>
34          <select id="MyMenu" size="3">
35              <option value="0">Toggle Rotation Direction</option>
36              <option value="1">Spin Faster</option>
37              <option value="2">Spin Slower</option>
38          </select>
39          <canvas id="gl-canvas" width="512" height="512">
40              Oops... your browser doesn't support the HTML5 canvas element!
41          </canvas>
42      </body>
```

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

rotateSquare.html | rotateSquare.js

C: > Users > sunje > Desktop > CG > rotateSquare.js > render

```
var gl;
var theta = 0;
var thetaLoc;
var direction = true;
var delay = 100;

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

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

    // Initialize event handlers
    document.getElementById("DirectionButton").onclick = function() {
        direction = !direction;
    }

    document.getElementById("MyMenu").onclick = function(event) {
        switch(event.target.index) {
            case 0:
                direction = !direction;
                break;
            case 1:
                delay *= 0.5;
                break;
            case 2:
                delay *= 2.0;
                break;
        }
    }

    // Four vertices
```

0 0 Ln 77, Col 13 Spaces: 4 UTF-8 CRLF JavaScript

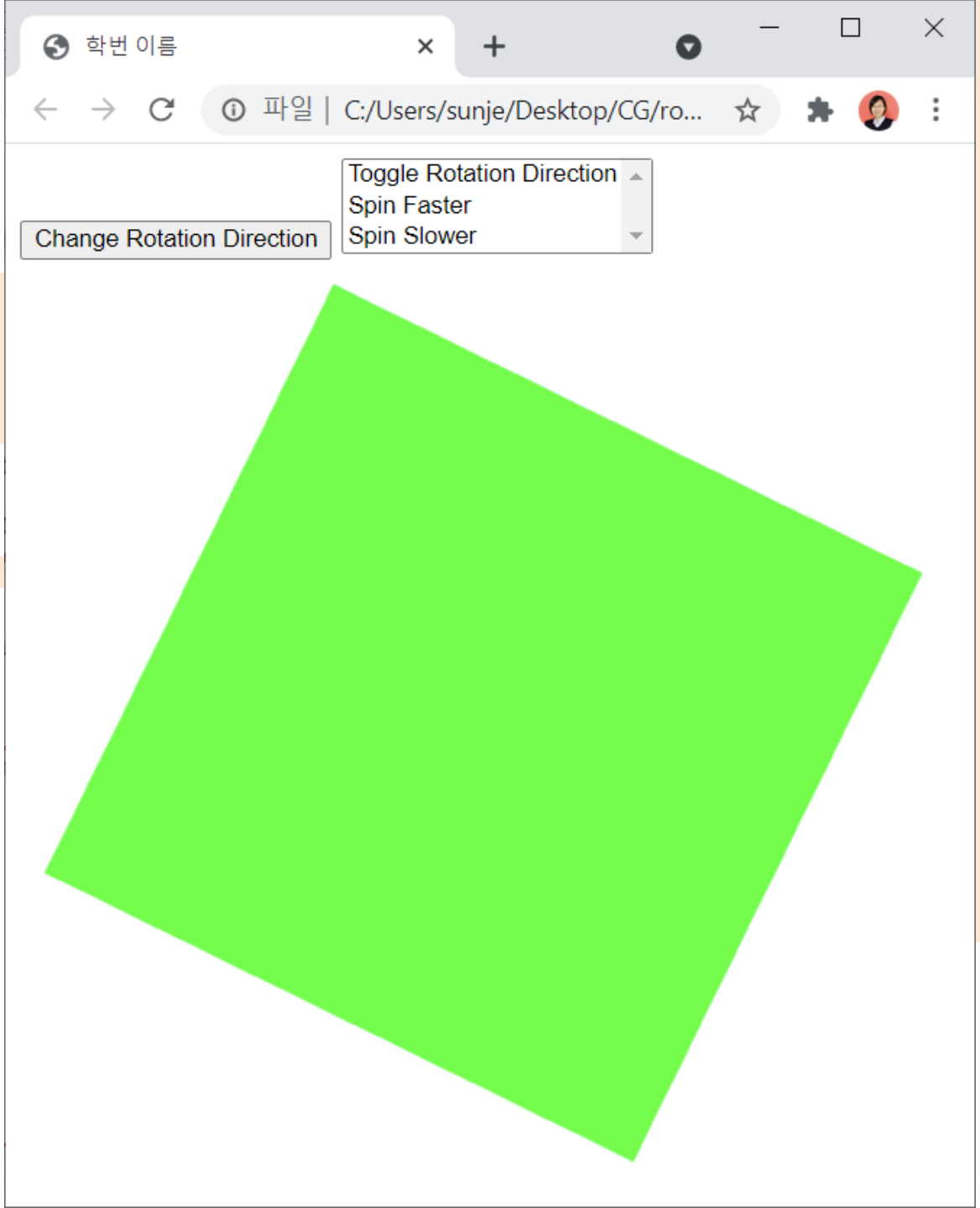
File Edit Selection View Go Run Terminal Help rotateSquare.js - Visual Studio Code

rotateSquare.html | rotateSquare.js

C: > Users > sunje > Desktop > CG > rotateSquare.js > render

```
47      // Load shaders and initialize attribute buffers
48      var program = initShaders(gl, "vertex-shader", "fragment-shader");
49      gl.useProgram(program);
50
51      // Load the data into the GPU
52      var bufferId = gl.createBuffer();
53      gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
54      gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);
55
56      // Associate our shader variables with our data buffer
57      var vPosition = gl.getAttribLocation(program, "vPosition");
58      gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
59      gl.enableVertexAttribArray(vPosition);
60
61      thetaLoc = gl.getUniformLocation(program, "theta");
62      //gl.uniform1f(thetaLoc, theta);
63
64      render();
65  };
66
67  function render() {
68      gl.clear(gl.COLOR_BUFFER_BIT);
69
70      theta += (direction ? 0.1 : -0.1);
71      gl.uniform1f(thetaLoc, theta);
72
73      gl.drawArrays(gl.TRIANGLE_STRIP, 0, 4);
74
75      setTimeout(function() {
76          window.requestAnimationFrame(render);
77      }, delay);
78  }
79
```

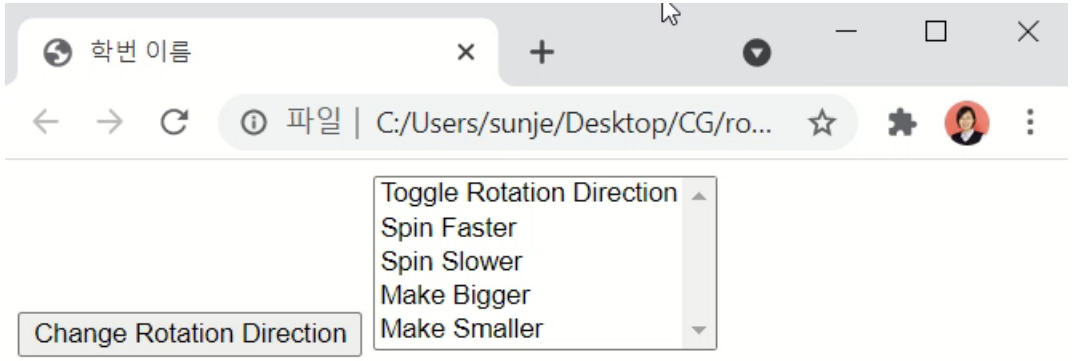
0 ⚠ 0 Ln 77, Col 13 Spaces: 4 UTF-8 CRLF JavaScript


학번 이름
파일 | C:/Users/sunje/Desktop/CG/ro...
Toggle Rotation Direction
Spin Faster
Spin Slower
Change Rotation Direction

index vs. value

```
21  document.getElementById("MyMenu").onclick = function(event) {
22      switch(event.target.value) {
23          case '0':
24              direction = !direction;
25              break;
26          case '1':
27              delay *= 0.5;
28              break;
29          case '2':
30              delay *= 2.0;
31              break;
32      }
33  }
```

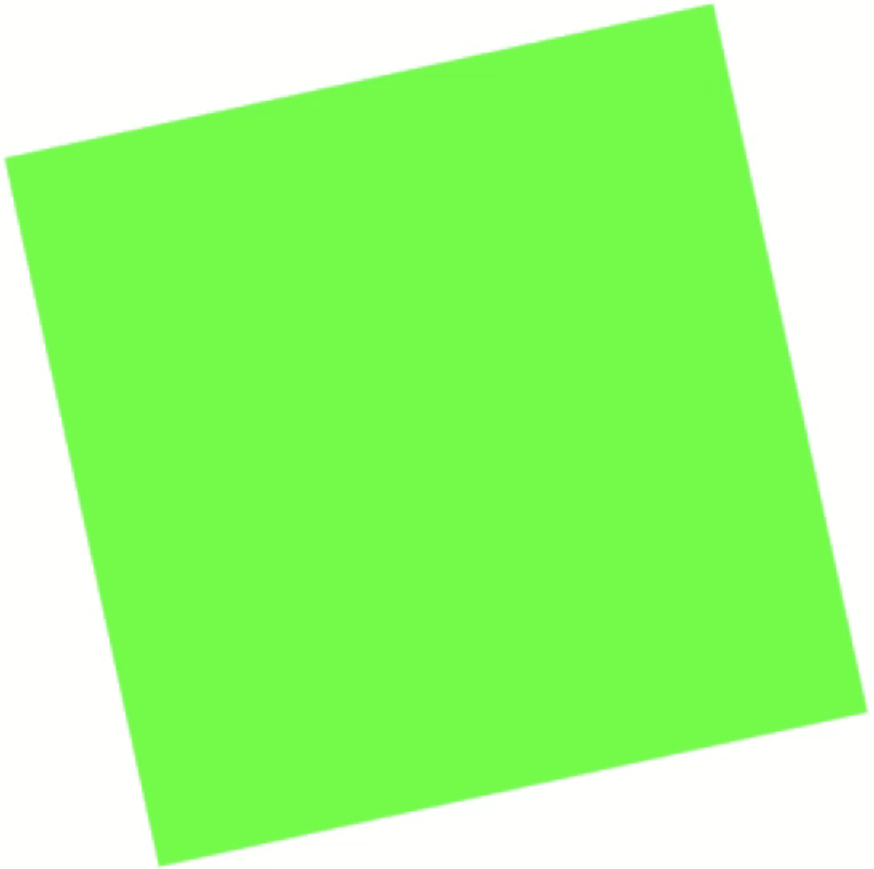
연습 문제 (3)



- 메뉴를 2개 더 추가하시오.
  - Make Bigger: length의 값을 1.1배 증가
  - Make Smaller: length의 값을 0.9배 감소

  - length는 uniform으로 vertex shader에 전달

  - float s = length * sin(theta);
  - float c = length * cos(theta);



Using “keydown” Event

```
window.addEventListener("keydown", function() {
    switch (event.keyCode) {
        case 49: // ’1’ key
            direction = !direction;
            break;
        case 50: // ’2’ key
            delay /= 2.0;
            break;
        case 51: // ’3’ key
            delay *= 2.0;
            break;
    }
});
```

Don’t know UNICODE

```
window.onkeydown = function(event) {
    var key = String.fromCharCode(event.keyCode);
    switch (key) {
      case ’1’:
        direction = !direction;
        break;
      case ’2’:
        delay /= 2.0;
        break;
      case ’3’:
        delay *= 2.0;
        break;
    }
};
```

Slider Element

- Puts slider on page
  - Give it an identifier
  - Give it minimum and maximum values
  - Give it a step size needed to generate an event
  - Give it an initial value
- Use div tag to put below canvas

```
41      <canvas id="gl-canvas" width="512" height="512">
42          Oops... your browser doesn't support the HTML5 canvas element!
43      </canvas>
44      <div>
45          Speed: 0 <input id="Slider" type="range" min="0" max="100" step="10" value="100"> 100
46      </div>
```

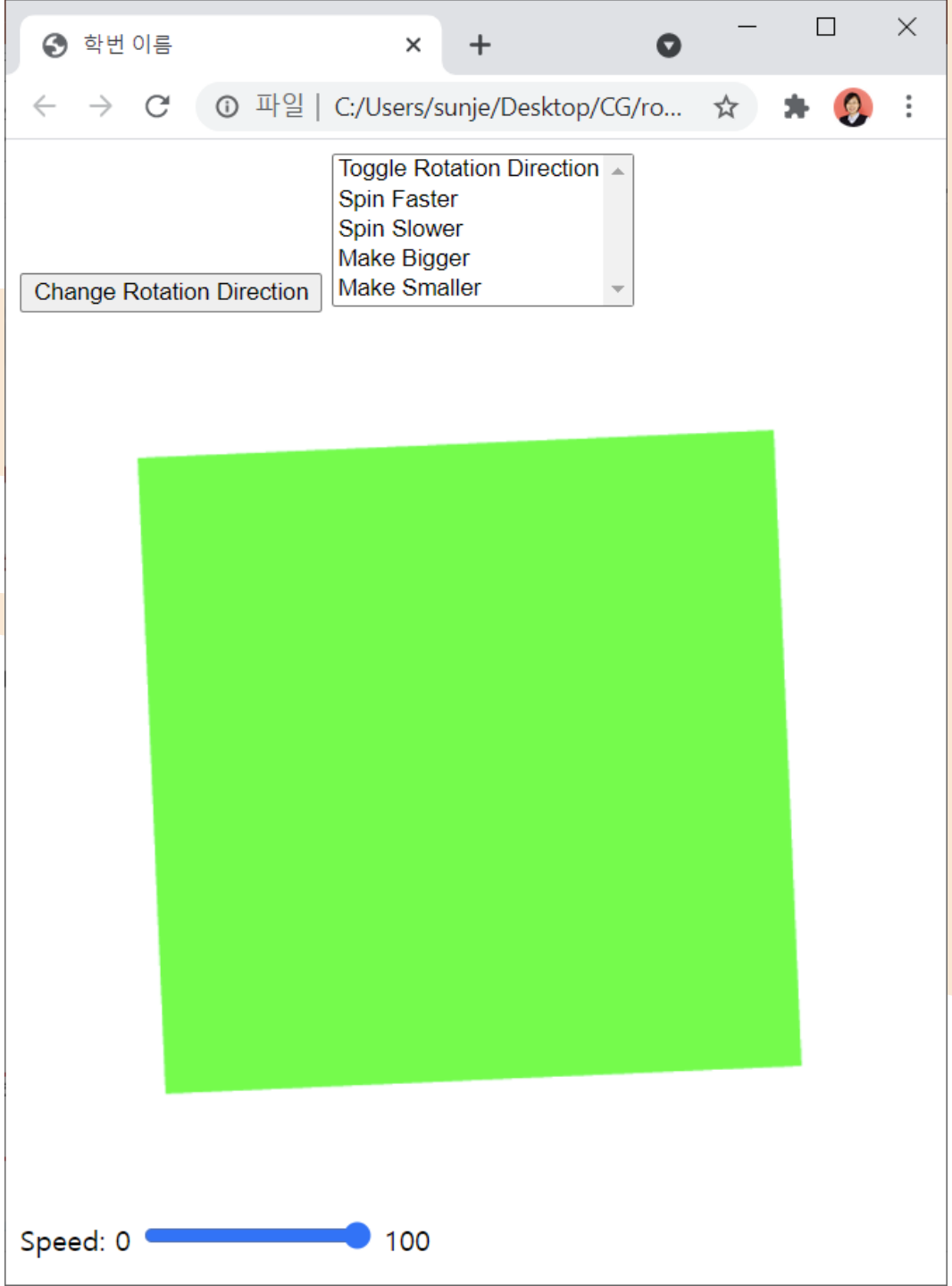
File Edit Selection View Go Run Terminal Help rotateSquare.js - Visual Studio Code

rotateSquare.html | rotateSquare.js

C: > Users > sunje > Desktop > CG > rotateSquare.js > init > onchange

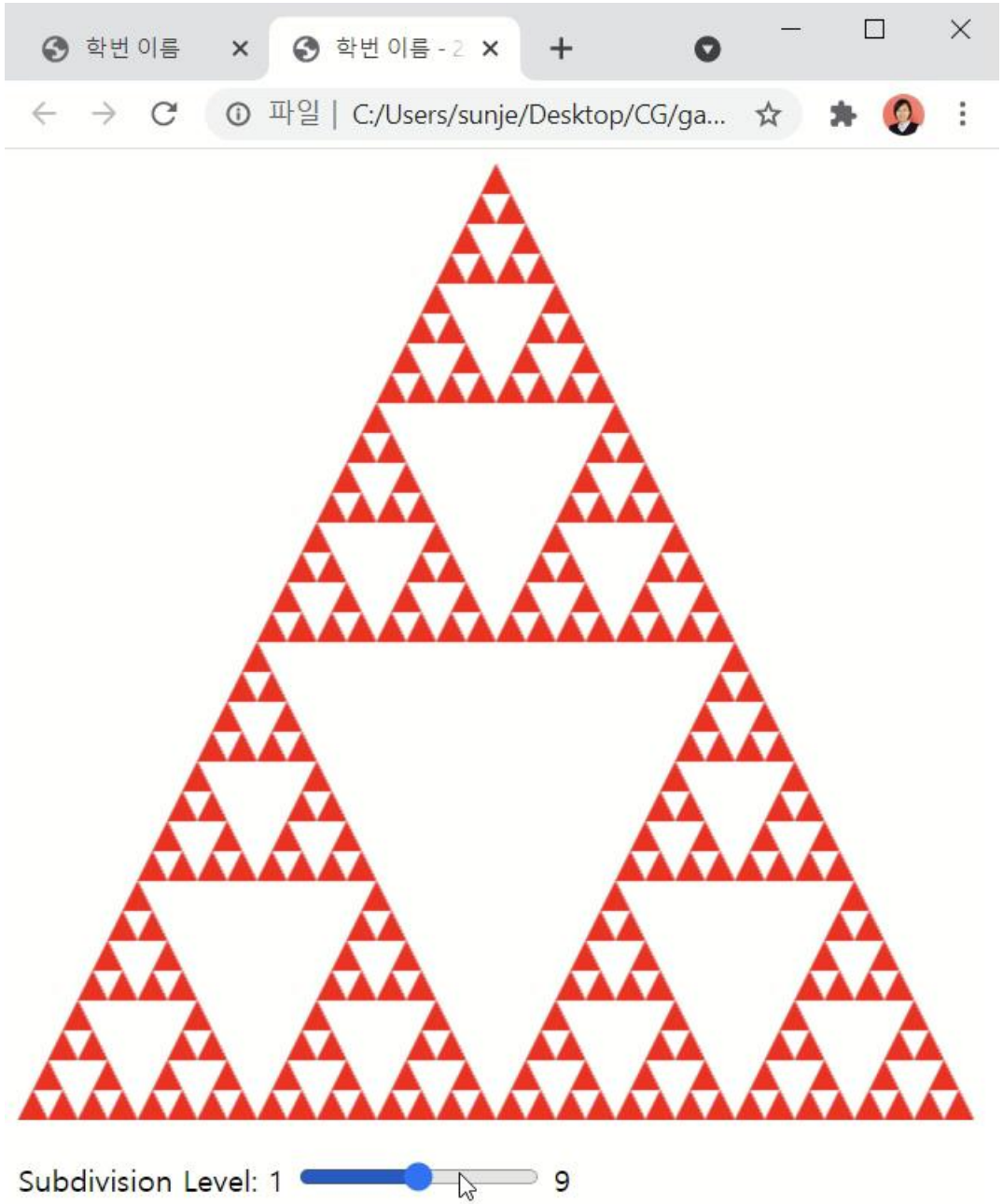
```
11      gl = WebGLUtils.setupWebGL(canvas);
12      if( !gl ) {
13          alert("WebGL isn't available!");
14      }
15
16      // Initialize event handlers
17      document.getElementById("DirectionButton").onclick = function() {
18          direction = !direction;
19      }
20
21      document.getElementById("MyMenu").onclick = function(event) {
22          switch(event.target.index) {
23              case 0:
24                  direction = !direction;
25                  break;
26              case 1:
27                  delay *= 0.5;
28                  break;
29              case 2:
30                  delay *= 2.0;
31                  break;
32              case 3:
33                  length *= 1.1;
34                  break;
35              case 4:
36                  length *= 0.9;
37                  break;
38          }
39      }
40
41      document.getElementById("Slider").onchange = function(event) {
42          delay = event.target.value;
43      }
44
```

0 ⚠ 0 Ln 42, Col 36 Spaces: 4 UTF-8 CRLF JavaScript


학번 이름
파일 | C:/Users/sunje/Desktop/CG/ro...
Toggle Rotation Direction
Spin Faster
Spin Slower
Make Bigger
Make Smaller
Change Rotation Direction
Speed: 0
100

연습 문제 (5)

- Recursive Subdivision 예제에서 Subdivision Level을 Slider로 조절할 수 있도록 구현하시오.



Window Events

- Events can be generated by actions that affect the canvas window
  - Moving or exposing a window
  - Resizing a window
  - Opening a window
  - Iconifying/deiconifying a window
- Note that events generated by other application that use the canvas can affect the WebGL canvas
  - There are default callbacks for some of these events

Reshape Events

- Suppose we use the mouse to change the size of our canvas
- Must redraw the contents
- Options
  - Display the same objects but change size
  - Display more or fewer objects at the same size
- Almost always want to keep proportions

“onresize” Event

- Returns size of new canvas is available through `window.innerHeight` and `window.innerWidth`
  - (`innerHeight`, `innerWidth`) ➔ (`canvas.height`, `canvas.width`)
- Ex) maintaining a square display

```
window.onresize = function() {
    var min = innerWidth;
    if (innerHeight < min) {
      min = innerHeight;
    }
    if (min < canvas.width || min < canvas.height) {
       gl.viewport(0, canvas.height-min, min, min);
    }
};
```