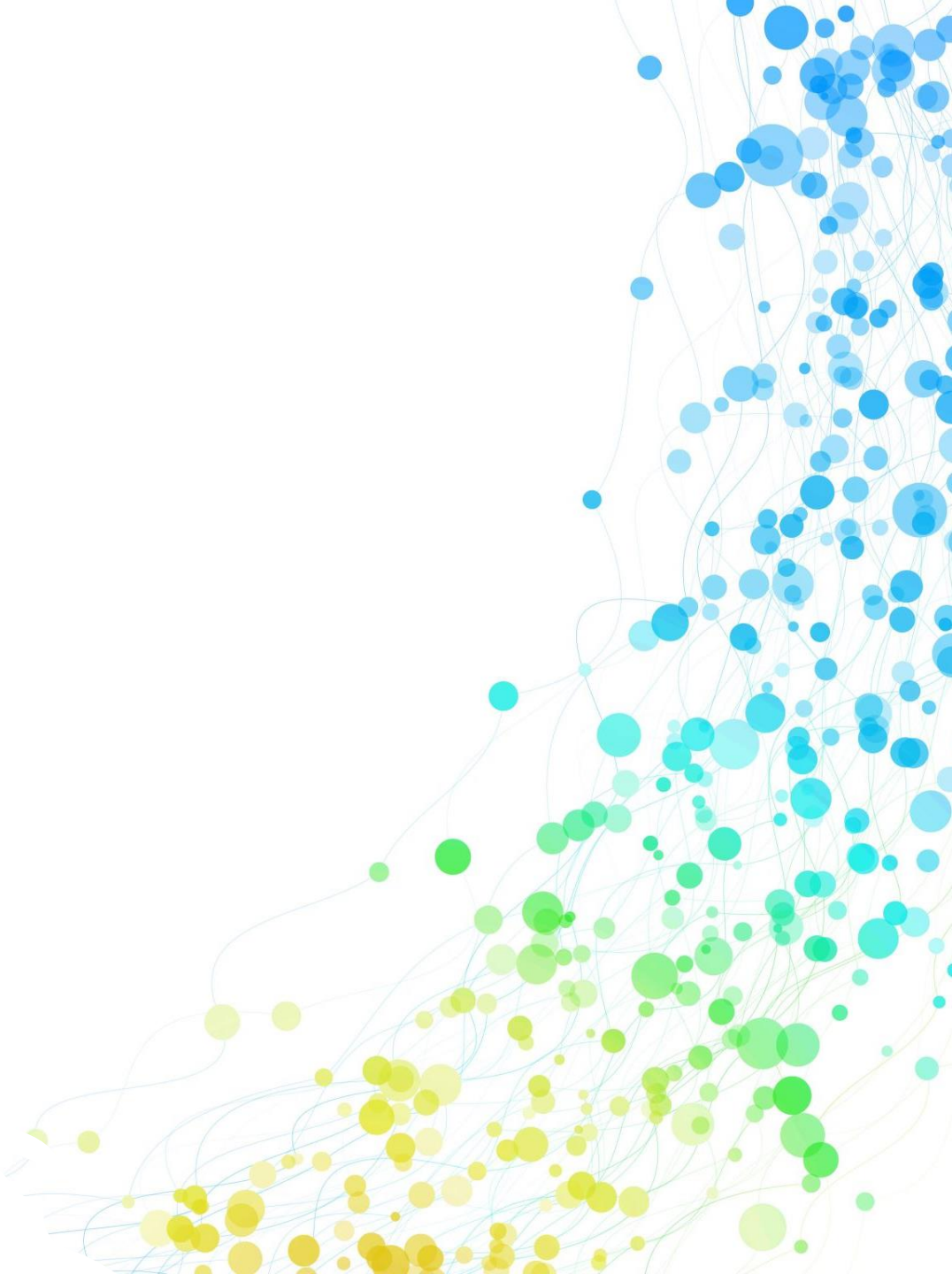
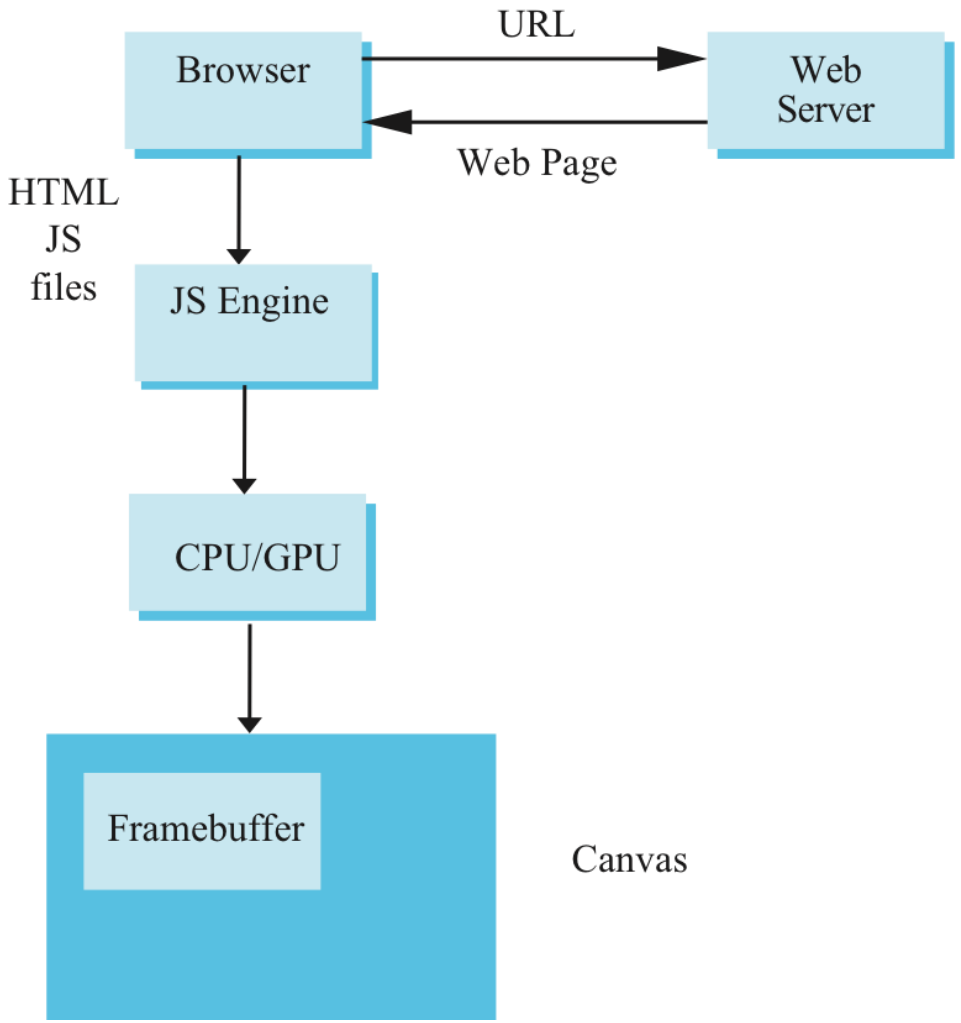
Interaction and Animation

4TH WEEK, 2022



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

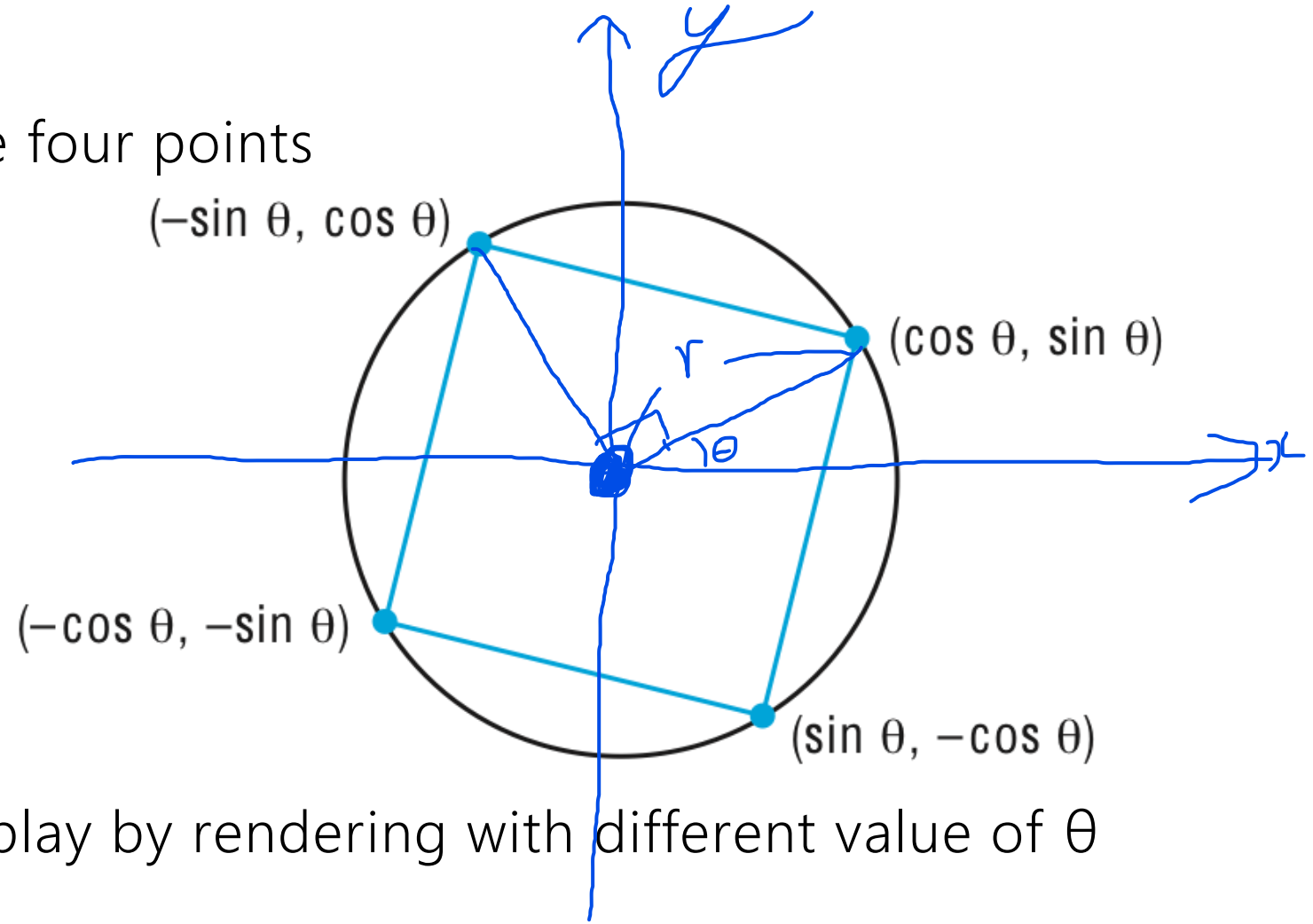
- Window: resize, expose, iconify
- Mouse: click one or more buttons
- Motion: move mouse
- Keyboard: press or release a key
- Idle: nonevent
  - Define what should be done if no other event is in queue

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



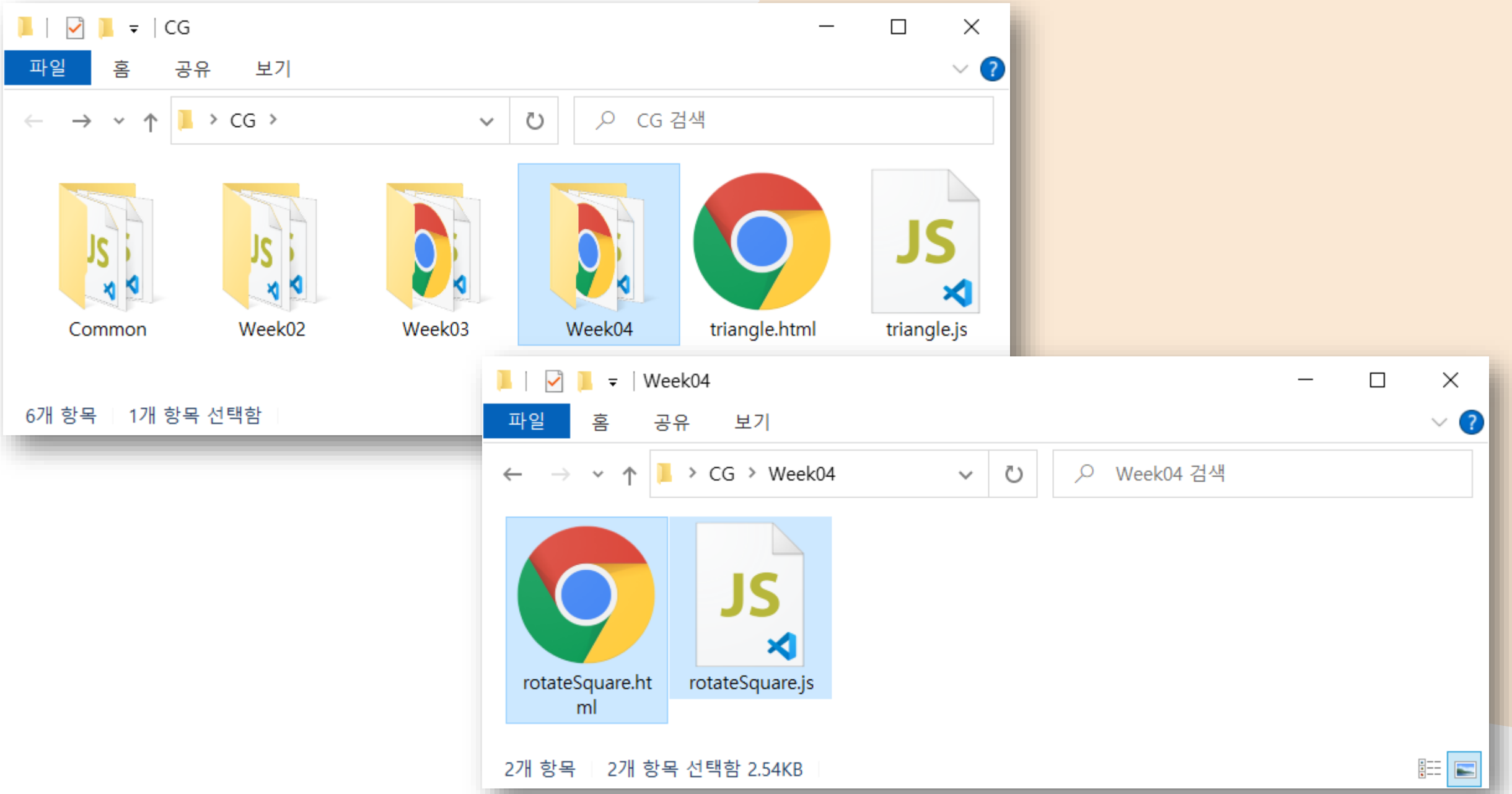
- Animate display by rendering with different value of θ

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


CG
파일
홈
공유
보기
CG
CG 검색
Common
Week02
Week03
Week04
triangle.html
triangle.js
6개 항목
1개 항목 선택함
Week04
파일
홈
공유
보기
CG
Week04
Week04 검색
rotateSquare.html
rotateSquare.js
2개 항목
2개 항목 선택함 2.54KB

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

rotateSquare.html | rotateSquare.js

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

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

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

        void main() {
            gl_Position = vPosition;
        }
        </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>
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 1, Col 1 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 > Sun-Jeong Kim > Desktop > CG > Week04 > rotateSquare.js > ...

```
1  var gl;
2
3  window.onload = function init()
4  {
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     var vertices = [
13         vec2(0, 1),
14         vec2(-1, 0),
15         vec2(0, -1),
16         vec2(1, 0)
17     ];
18
19     // Configure WebGL
20     gl.viewport(0, 0, canvas.width, canvas.height);
21     gl.clearColor(1.0, 1.0, 1.0, 1.0);
22
23     // Load shaders and initialize attribute buffers
24     var program = initShaders(gl, "vertex-shader", "fragment-shader");
25     gl.useProgram(program);
26
27     // Load the data into the GPU
28     var bufferId = gl.createBuffer();
29     gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
30     gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);
31
32     // Associate our shader variables with our data buffer
33     var vPosition = gl.getAttribLocation(program, "vPosition");
34     gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
35     gl.enableVertexAttribArray(vPosition);
```

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

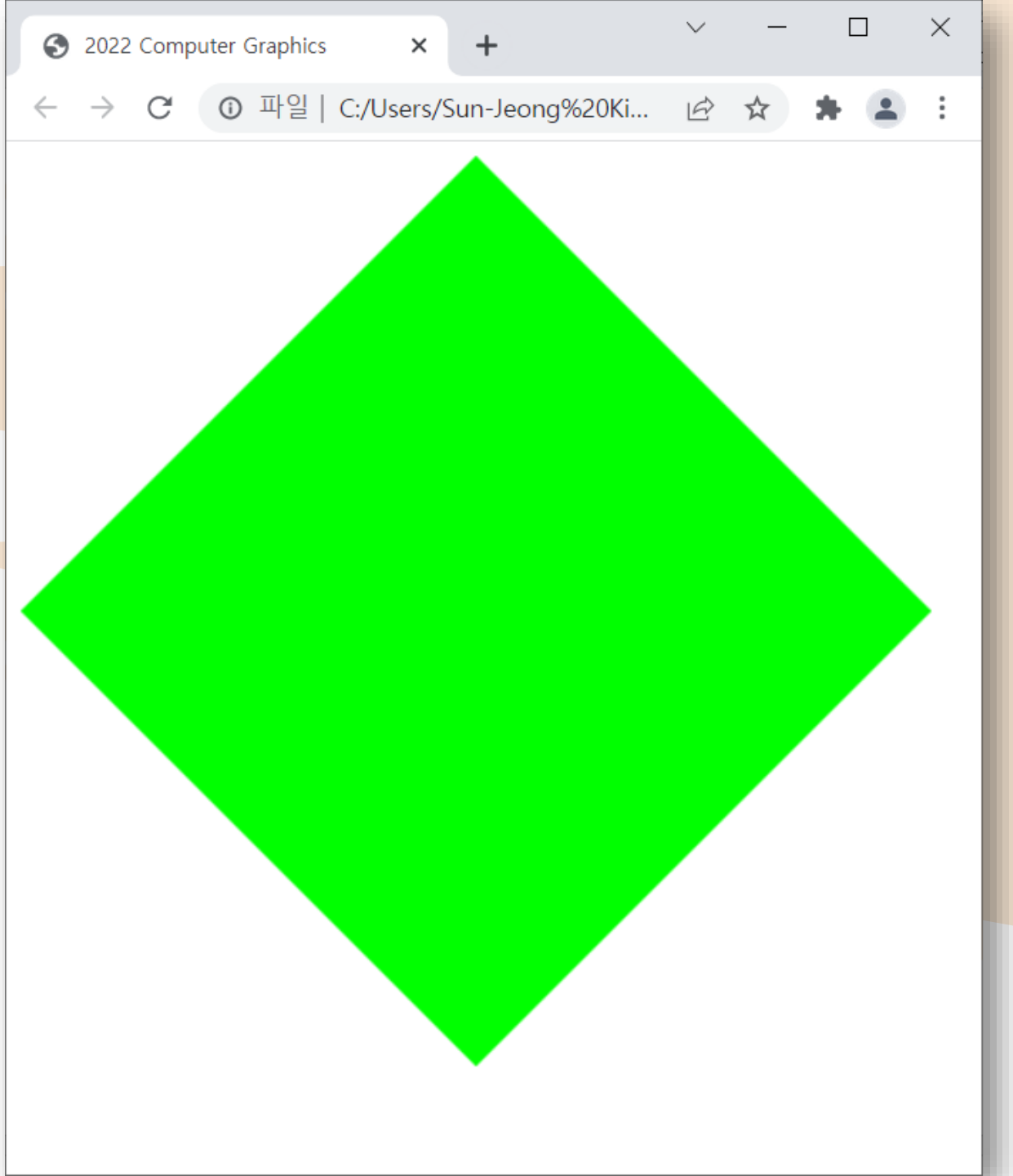
File Edit Selection View Go Run Terminal Help rotateSquare.js - Visual Studio Code

rotateSquare.html | rotateSquare.js

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

```
12      var vertices = [
13          vec2(0, 1),
14          vec2(-1, 0),
15          vec2(0, -1),
16          vec2(1, 0)
17      ];
18
19      // Configure WebGL
20      gl.viewport(0, 0, canvas.width, canvas.height);
21      gl.clearColor(1.0, 1.0, 1.0, 1.0);
22
23      // Load shaders and initialize attribute buffers
24      var program = initShaders(gl, "vertex-shader", "fragment-shader");
25      gl.useProgram(program);
26
27      // Load the data into the GPU
28      var bufferId = gl.createBuffer();
29      gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
30      gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);
31
32      // Associate our shader variables with our data buffer
33      var vPosition = gl.getAttribLocation(program, "vPosition");
34      gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
35      gl.enableVertexAttribArray(vPosition);
36
37      render();
38  };
39
40  function render()
41  {
42      gl.clear(gl.COLOR_BUFFER_BIT);
43      gl.drawArrays(gl.TRIANGLE_FAN, 0, 4);
44  }
45
```

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


2022 Computer Graphics
파일 | C:/Users/Sun-Jeong%20Ki...

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

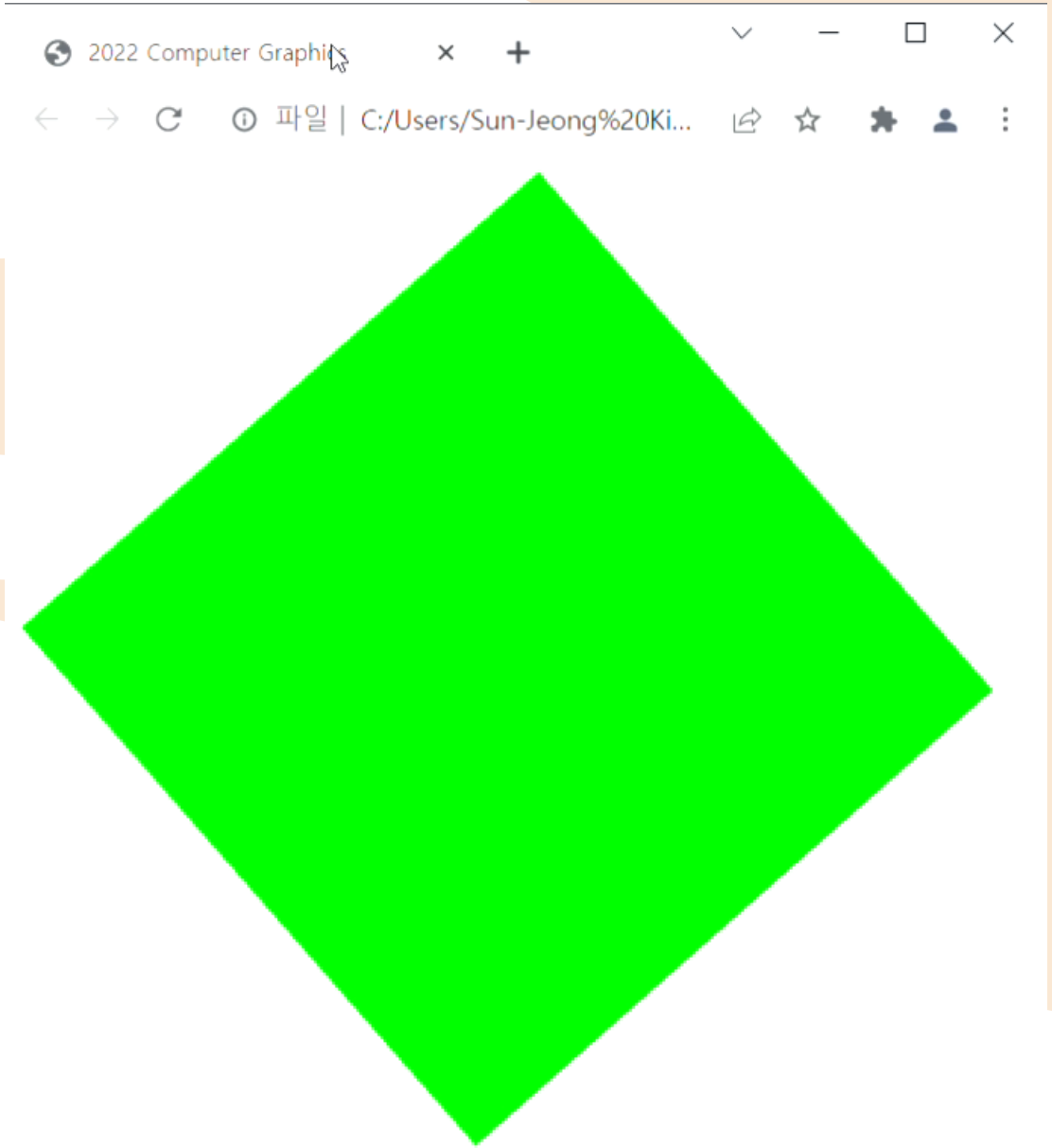
- Executes a function after a specified number of milliseconds
  - Also generates a buffer swap

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

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

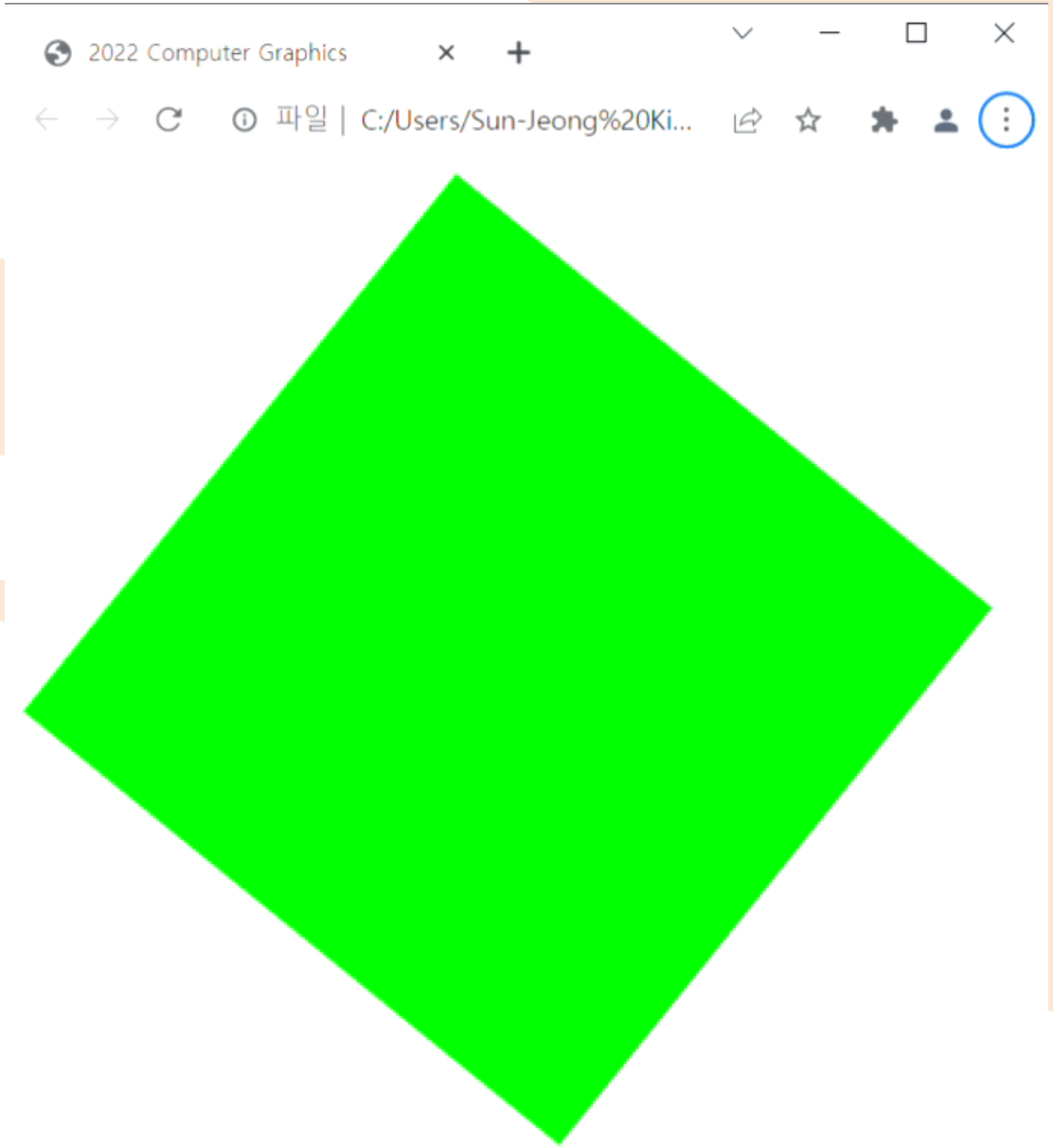
requestAnimFrame

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


2022 Computer Graphics
파일 | C:/Users/Sun-Jeong%20Ki...

Add an Interval

```
45  function render()
46  {
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_FAN, 0, 4);
54
55          window.requestAnimationFrame(render);
56      }, 100);
57  }
```


2022 Computer Graphics
파일 | C:/Users/Sun-Jeong%20Ki...

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

rotateSquare.html rotateSquare.js

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

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

Restricted Mode 0 0 Ln 16, Col 32 Spaces: 4 UTF-8 CRLF HTML

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

rotateSquare.html | rotateSquare.js

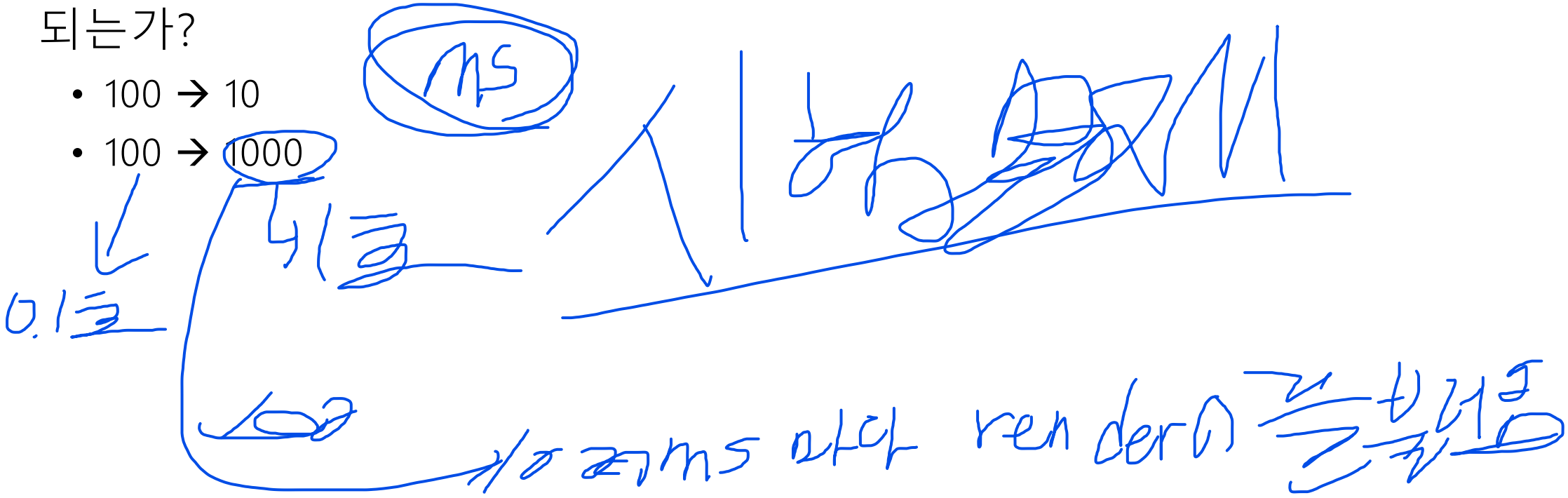
C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > rotateSquare.js > render

```
25      // Load shaders and initialize attribute buffers
26      var program = initShaders(gl, "vertex-shader", "fragment-shader");
27      gl.useProgram(program);
28
29      // Load the data into the GPU
30      var bufferId = gl.createBuffer();
31      gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
32      gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);
33
34      // Associate our shader variables with our data buffer
35      var vPosition = gl.getAttribLocation(program, "vPosition");
36      gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
37      gl.enableVertexAttribArray(vPosition);
38
39      thetaLoc = gl.getUniformLocation(program, "theta");
40      gl.uniform1f(thetaLoc, theta);
41
42      render();
43  };
44
45  function render()
46  {
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_FAN, 0, 4);
54
55          window.requestAnimationFrame(render);
56      }, 100);
57  }
58
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 56, Col 11 Spaces: 4 UTF-8 CRLF {} JavaScript

연습문제 (1)

- SetTimeout( ) 함수의 입력 파라미터를 아래와 같이 변경하면, 어떻게 되는가?
  - 100 → 10
  - 100 → 1000



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

1.

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

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

2.

```
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 > Sun-Jeong Kim > Desktop > CG > Week04 > 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>
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 34, 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 > Sun-Jeong Kim > Desktop > CG > Week04 > rotateSquare.js > render > setTimeout() callback

```
1   var gl;
2   var theta = 0;
3   var thetaLoc;
4   var direction = true;
5
6   window.onload = function init()
7   {
8       var canvas = document.getElementById("gl-canvas");
9
10      gl = WebGLUtils.setupWebGL(canvas);
11      if( !gl ) {
12          alert("WebGL isn't available!");
13      }
14
15      // Initialize event handlers
16      document.getElementById("directionButton").onclick = function() {
17          direction = !direction;
18      };
19
20      var vertices = [
21          vec2(0, 1),
22          vec2(-1, 0),
23          vec2(0, -1),
24          vec2(1, 0)
25      ];
26
27      // Configure WebGL
28      gl.viewport(0, 0, canvas.width, canvas.height);
29      gl.clearColor(1.0, 1.0, 1.0, 1.0);
30
31      // Load shaders and initialize attribute buffers
32      var program = initShaders(gl, "vertex-shader", "fragment-shader");
33      gl.useProgram(program);
34
35      // Load the data into the GPU
```

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

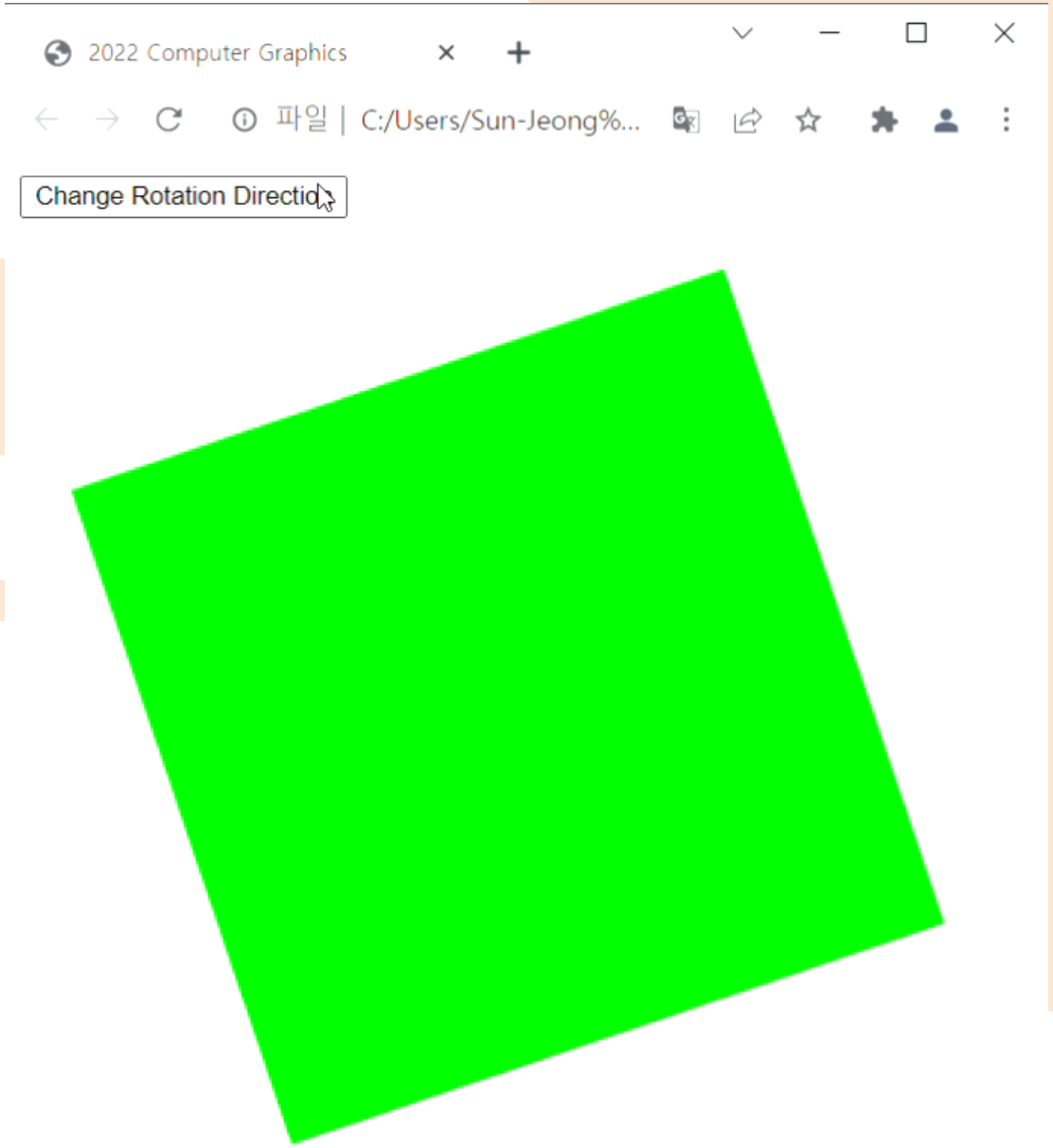
File Edit Selection View Go Run Terminal Help rotateSquare.js - Visual Studio Code

rotateSquare.html | rotateSquare.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week04 > rotateSquare.js > render > setTimeout() callback

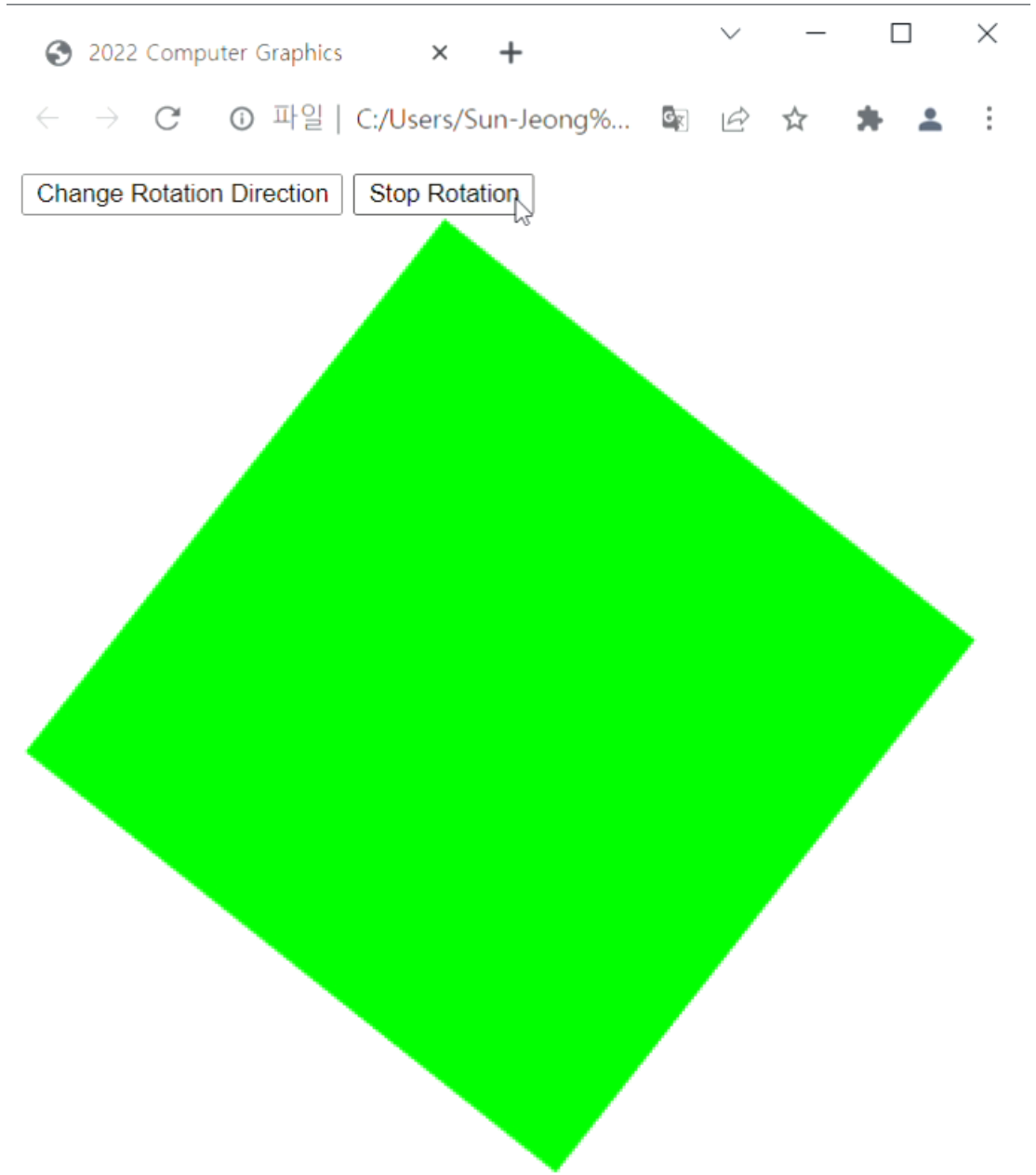
```
31      // Load shaders and initialize attribute buffers
32      var program = initShaders(gl, "vertex-shader", "fragment-shader");
33      gl.useProgram(program);
34
35      // Load the data into the GPU
36      var bufferId = gl.createBuffer();
37      gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);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

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


2022 Computer Graphics
파일 | C:/Users/Sun-Jeong%...
Change Rotation Directio

연습문제 (2)

- 회전을 멈추거나 시작하는 Toggle 버튼을 추가하시오.



Menus

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

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

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

rotateSquare.html | rotateSquare.js

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

```
1   var gl;
2   var theta = 0;
3   var thetaLoc;
4   var direction = true;
5   var stop = false;
6   var delay = 100;
7
8   window.onload = function init()
9   {
10      var canvas = document.getElementById("gl-canvas");
11
12      gl = WebGLUtils.setupWebGL(canvas);
13      if( !gl ) {
14          alert("WebGL isn't available!");
15      }
16
17      // Initialize event handlers
18      document.getElementById("directionButton").onclick = function() {
19          direction = !direction;
20      };
21      document.getElementById("myMenu").onclick = function(event) {
22          switch(event.target.index) {
23              case 0:
24                  delay *= 0.5;
25                  break;
26              case 1:
27                  delay *= 2.0;
28                  break;
29          }
30      };
31      document.getElementById("stopButton").onclick = function(event) {
32          stop = !stop;
33          if( stop ) {
34              event.target.innerText = "Start Rotation";
35          }
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 28, Col 23 Spaces: 4 UTF-8 CRLF {} JavaScript

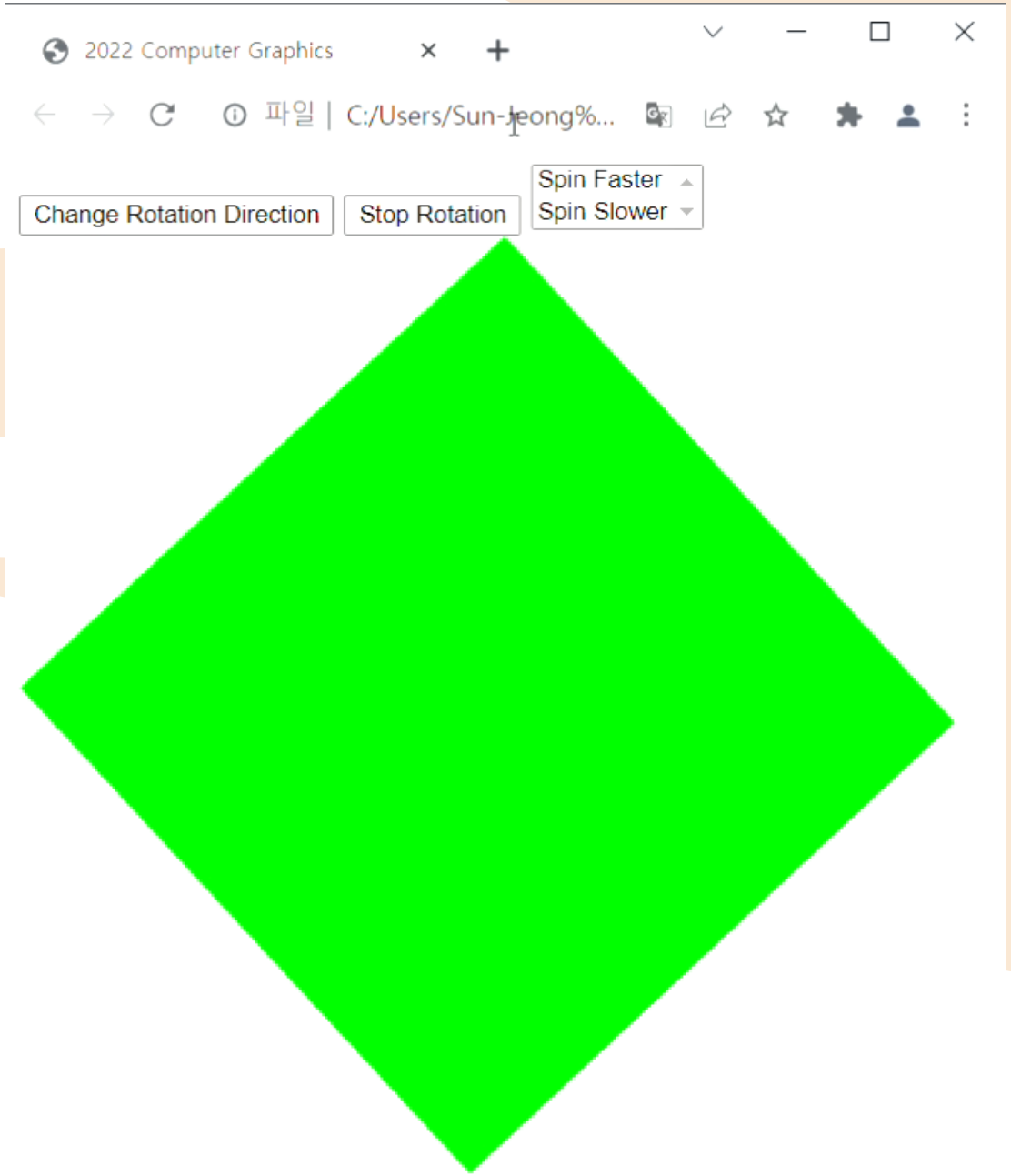
File Edit Selection View Go Run Terminal Help rotateSquare.js - Visual Studio Code

rotateSquare.html | rotateSquare.js

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

```
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(vertices), 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      thetaLoc = gl.getUniformLocation(program, "theta");
67      gl.uniform1f(thetaLoc, theta);
68
69      render();
70  };
71
72  function render()
73  {
74      setTimeout(function() {
75          gl.clear(gl.COLOR_BUFFER_BIT);
76
77          theta += (stop ? 0 : (direction ? 0.1 : -0.1));
78          gl.uniform1f(thetaLoc, theta);
79
80          gl.drawArrays(gl.TRIANGLE_FAN, 0, 4);
81
82          window.requestAnimationFrame(render);
83      }, delay);
84  }
85
```

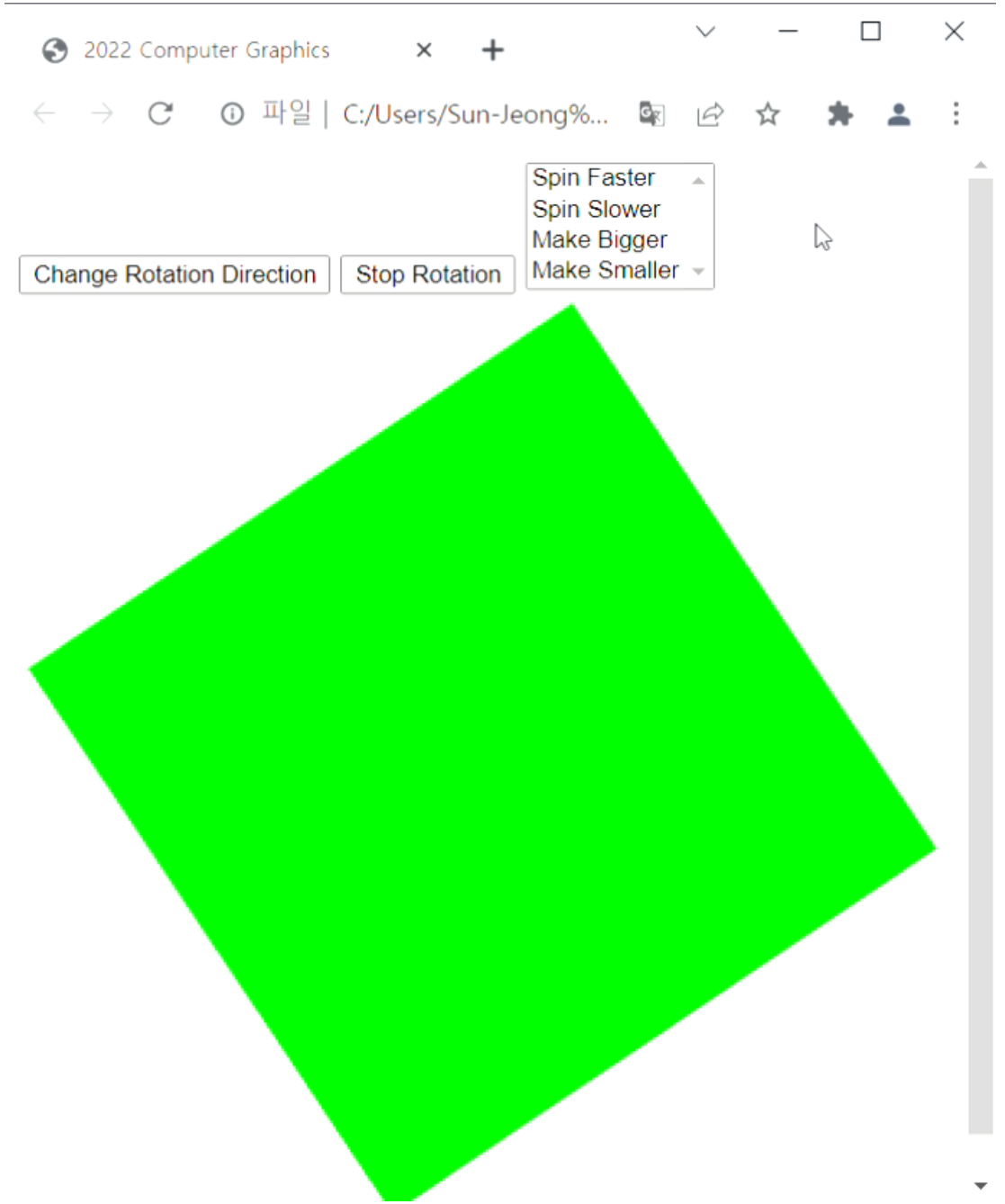
Restricted Mode ⊗ 0 ⚠ 0 Ln 83, Col 13 Spaces: 4 UTF-8 CRLF {} JavaScript


2022 Computer Graphics
파일 | C:/Users/Sun-Jeong%...
Change Rotation Direction
Stop Rotation
Spin Faster
Spin Slower

index vs. value

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

연습문제 (3)



- 메뉴를 2개 더 추가하시오.
  - Make Bigger: length의 값을 1.1배 증가
  - Make Smaller: length의 값을 0.9배 감소
  - length는 uniform으로 vertex shader에 전달 (theta 전달 방식 참조)
  - 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

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

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

rotateSquare.html | rotateSquare.js

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

```
7   var length = 1.0;
8   var lengthLoc;
9
10  window.onload = function init()
11  {
12      var canvas = document.getElementById("gl-canvas");
13
14      gl = WebGLUtils.setupWebGL(canvas);
15      if( !gl ) {
16          alert("WebGL isn't available!");
17      }
18
19      // Initialize event handlers
20      document.getElementById("directionButton").onclick = function() {
21          direction = !direction;
22      };
23      document.getElementById("myMenu").onclick = function(event) {
24          switch(event.target.value) {
25              case '0':
26                  delay *= 0.5;
27                  break;
28              case '1':
29                  delay *= 2.0;
30                  break;
31              case '2':
32                  length *= 1.1;
33                  break;
34              case '3':
35                  length *= 0.9;
36                  break;
37          }
38      };
39      document.getElementById("speedSlider").onchange = function(event) {
40          delay = event.target.value;
41      };
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 40, Col 36 Spaces: 4 UTF-8 CRLF {} JavaScript


2022 Computer Graphics
파일 | C:/Users/Sun-Jeong%...
Spin Faster
Spin Slower
Make Bigger
Make Smaller
Change Rotation Direction
Stop Rotation
Rotation Speed: 10
100

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);
    }
};
```