**컴퓨터 그래픽스(SW) 과제 1**

**:** **Sierpinski Carpet**



**학번: 32193430**

**이름: 이재원**

**담당교수: 송 인 식 교수님**

**분반: 2분반**

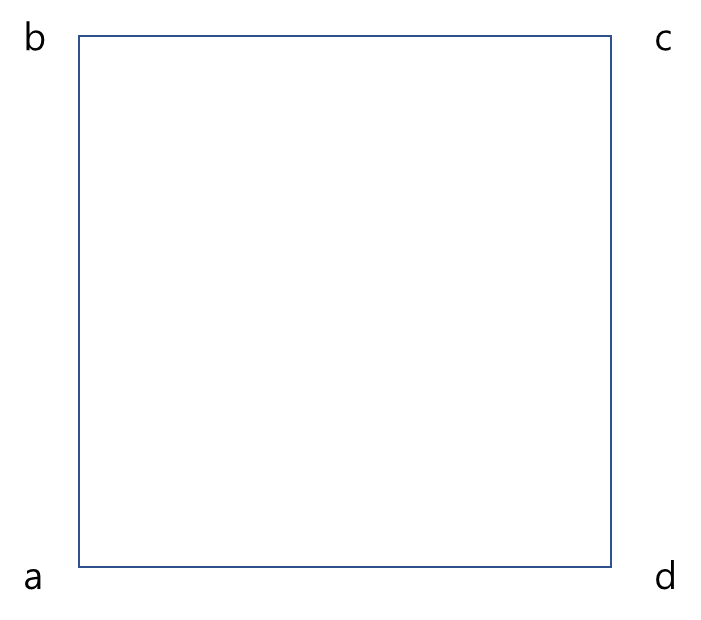
**제출일: 2021. 09. 28**

코드 편집은 Visual Studio Code 소프트웨어를 이용하였다.

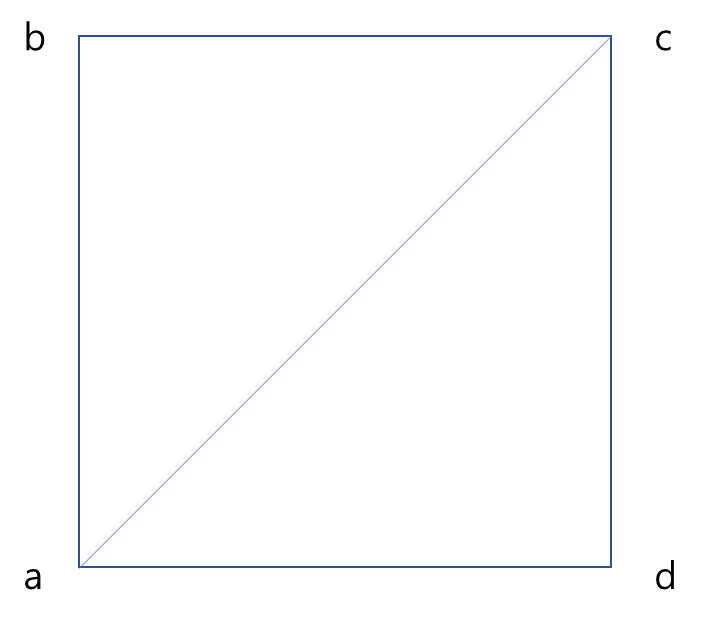
1. **핵심 아이디어**

WebGL은 삼각형만 그릴 수 있다. 따라서 다각형을 구현할 때는 삼각형으로 분할하여 그려야 한다.

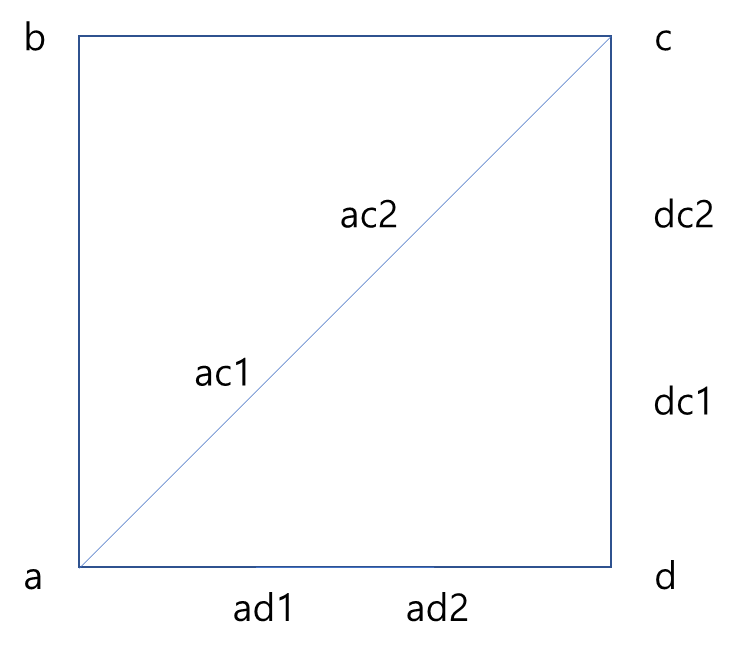
1. 다음과 같이 네 점 a, b, c, d를 연결하여 사각형을 그린다.



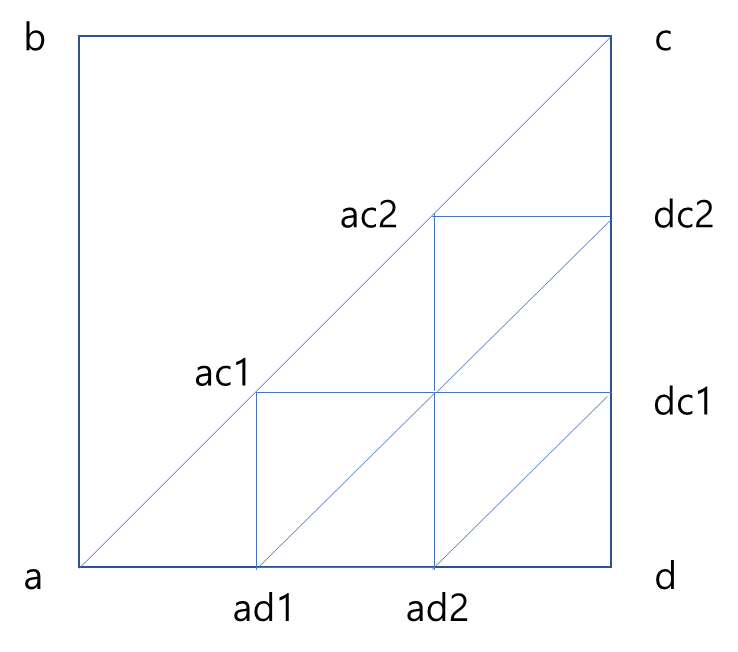
1. a와 c를 연결하여 두 개의 삼각형으로 분할한다.



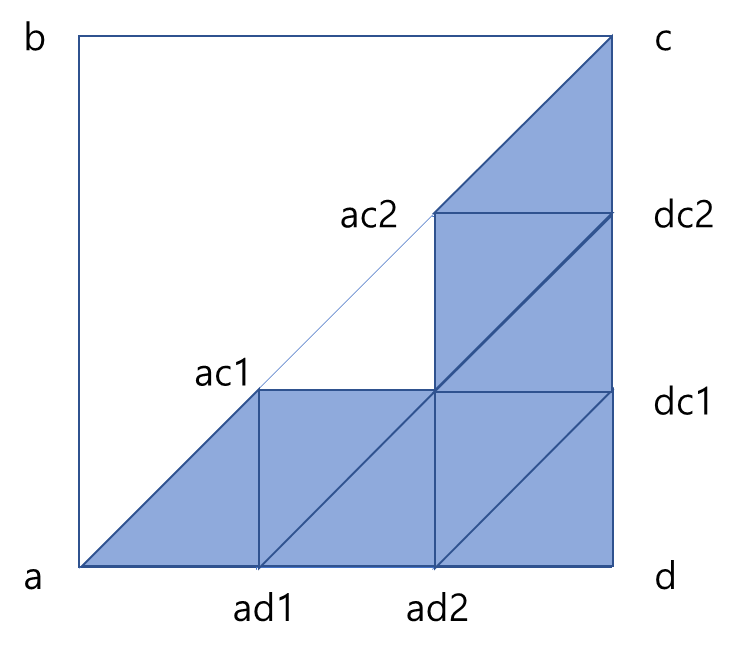
1. 변 ac, ad, dc를 각각 3등분한다.



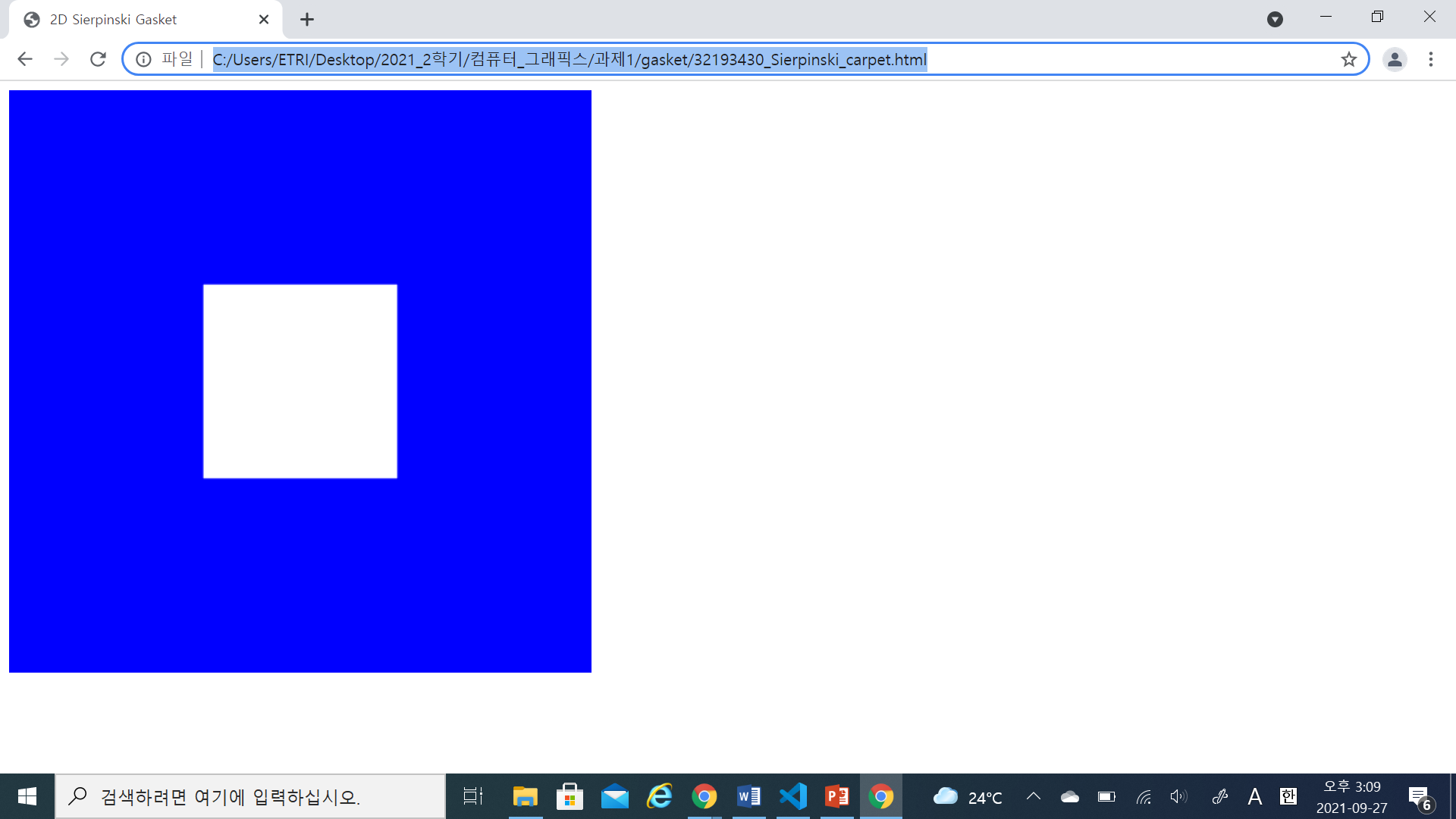
1. 각 점을 연결하여 9개의 삼각형으로 분할한다.



1. ac1 – ac2 변을 포함하는 삼각형을 제외하고 나머지 삼각형을 구현한다.

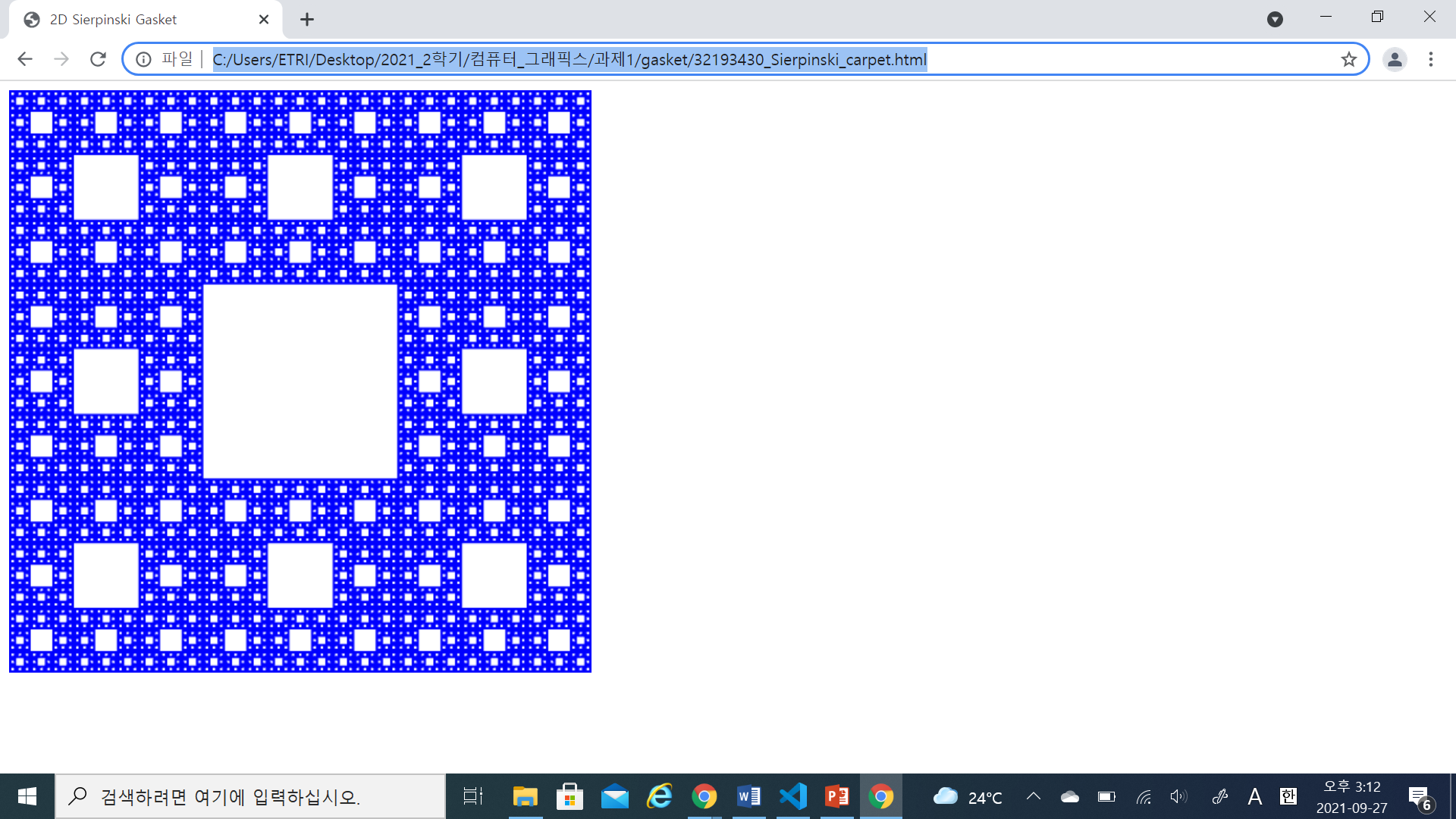


1. 삼각형 abc에 1~5의 과정을 똑같이 반복한다.



실제 WebGL로 구현한 모습

1. 4에서 분할한 삼각형에 1~6과정을 원하는 횟수만큼 반복한다.



1~6 과정을 5번 반복한 결과: **Sierpinski Carpet**

1. **주요 코드 설명**

* **HTML**

1. **Square의 색과 투명도**

<script id="fragment-shader" type="x-shader/x-fragment">

precision mediump float;

void

main()

{

    gl\_FragColor = vec4( 0.0, 0.0, 1.0, 1.0 );

}

</script>

<!--색깔과 투명도를 나타낸다!-->

1. **JavaScript 파일 읽어 들이기**

<script type="text/javascript" src="C:\Users\ETRI\Desktop\2021\_2학기\컴퓨터\_그래픽스\과제1\32193430\_컴퓨터\_그래픽스\_과제 1\_Sierpinski\_Carpet\webgl-utils.js"></script>

<script type="text/javascript" src="C:\Users\ETRI\Desktop\2021\_2학기\컴퓨터\_그래픽스\과제1\32193430\_컴퓨터\_그래픽스\_과제 1\_Sierpinski\_Carpet\initShaders.js"></script>

<script type="text/javascript" src="C:\Users\ETRI\Desktop\2021\_2학기\컴퓨터\_그래픽스\과제1\32193430\_컴퓨터\_그래픽스\_과제 1\_Sierpinski\_Carpet\MV.js"></script>

<script type="text/javascript" src="32193430\_이재원\_Sierpinski\_carpet(최종).js"></script>

* **JavaScript**

1. **변수 선언**

"use strict";

var canvas;

var gl;

var points = [];

var NumTimesToSubdivide = 5; // 분할을 반복하고 싶은 횟수(크기가 서로 다른 사각형의 개수), 최대 6까지 가능!

1. **사각형 그리기**

var vertices = [

vec2( -1, -1 ), // a

vec2( -1,  1 ), // b

vec2( 1, 1 ), // c

vec2( 1, -1 ) // d ];

1. **Points 배열에 각 Vertices 삽입**

function triangle( a, b, c )

{

points.push( a, b, c );

}

1. **삼각형 분할 재귀 함수 divideTriangle 구현하기**

function divideTriangle( a, d, c, count ) //a, d, c는 각 Vertices, count는 반복 횟수

{

  // check for end of recursion

    if ( count === 0 ) { // 반복횟수가 0이면 삼각형 그리기

        triangle( a, d, c );

    }

    else {

        //변을 삼 등분하기, 7개의 Vertices 추가 생성

        var ad1 = mix( a, d, 1/3 );

        var ad2 = mix( a, d, 2/3 );

       var dc1 = mix( d, c, 1/3 );

        var dc2 = mix( d, c, 2/3 );

        var ac1 = mix( a, c, 1/3 );

        var ac2 = mix( ac1, c, 1/2 );

        var h = mix(ad1, dc2, 1/2);

        --count;

 // 8개의 새로운 삼각형 생성, 삼각형 분할을 count가 0이 될 때까지 반복

        divideTriangle( a, ad1, ac1, count );

        divideTriangle( ad1, ac1, h, count );

        divideTriangle( ad1, ad2, h, count );

        divideTriangle( ad2, d, dc1, count );

        divideTriangle( ad2, h, dc1, count );

        divideTriangle( h, dc1, dc2, count );

        divideTriangle( h, ac2, dc2, count );

        divideTriangle( ac2, dc2, c, count );

    }

}

1. **render() 함수 구현하기**

function render()

{

gl.clear( gl.COLOR\_BUFFER\_BIT );

gl.drawArrays( gl.TRIANGLES, 0, points.length ); // gl.TRIANGLES 사용!

}

1. **삼각형 2개 그려 Sierpinski Carpet 구현하기**

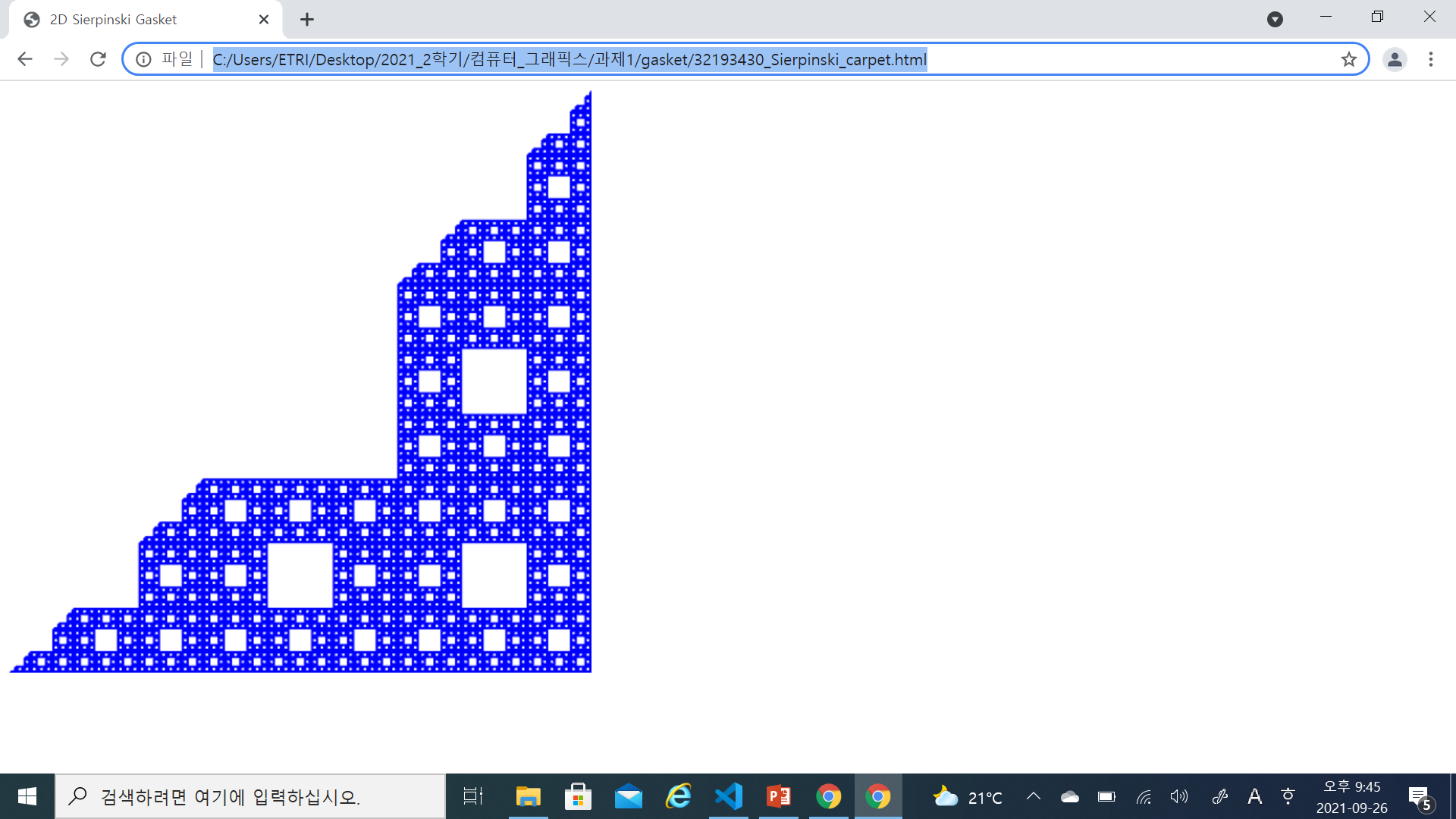
divideTriangle( vertices[0], vertices[3], vertices[2],

  NumTimesToSubdivide); // vertices가 다른 것에 유의!

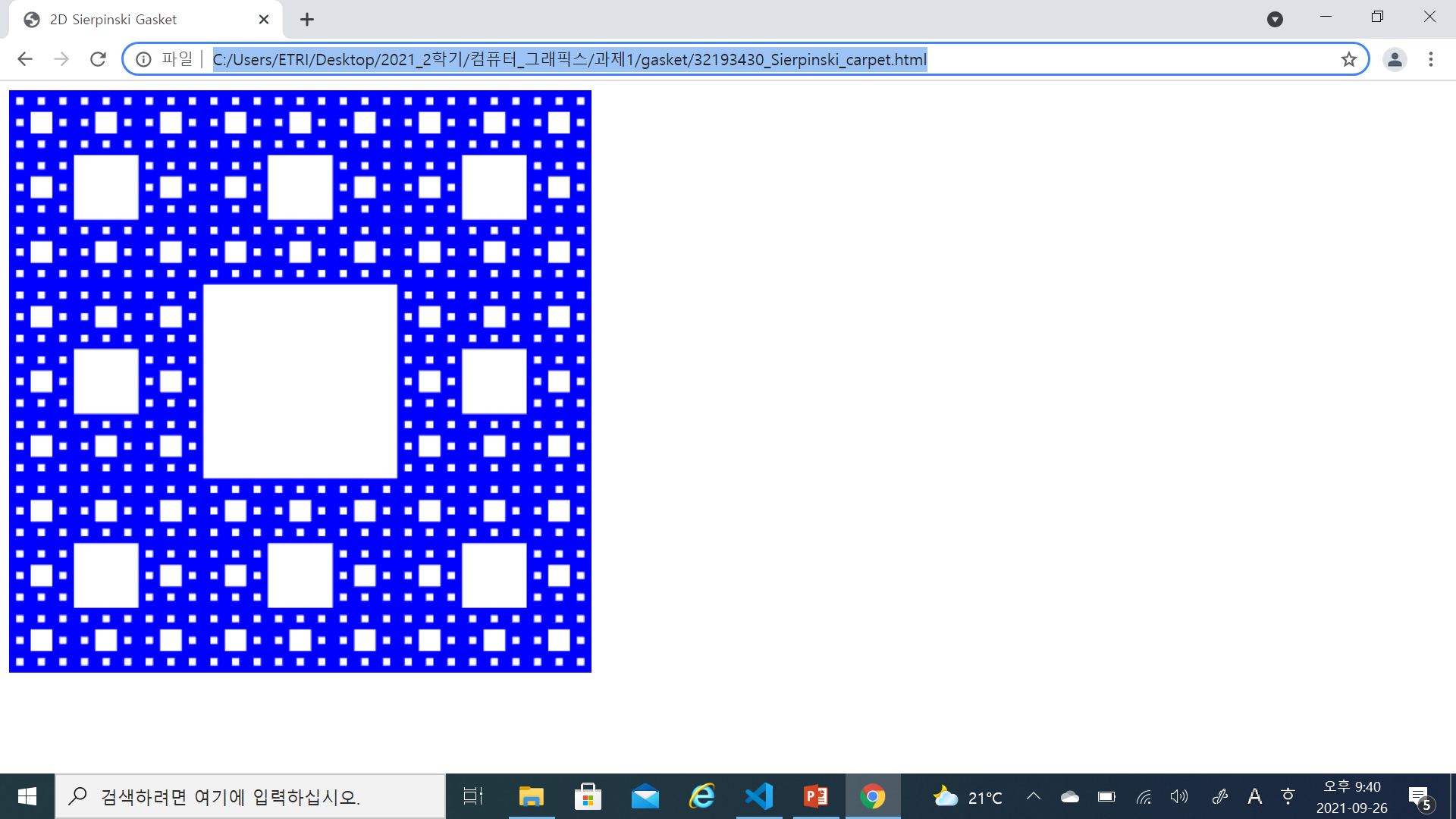
divideTriangle( vertices[0], vertices[1], vertices[2],

NumTimesToSubdivide);

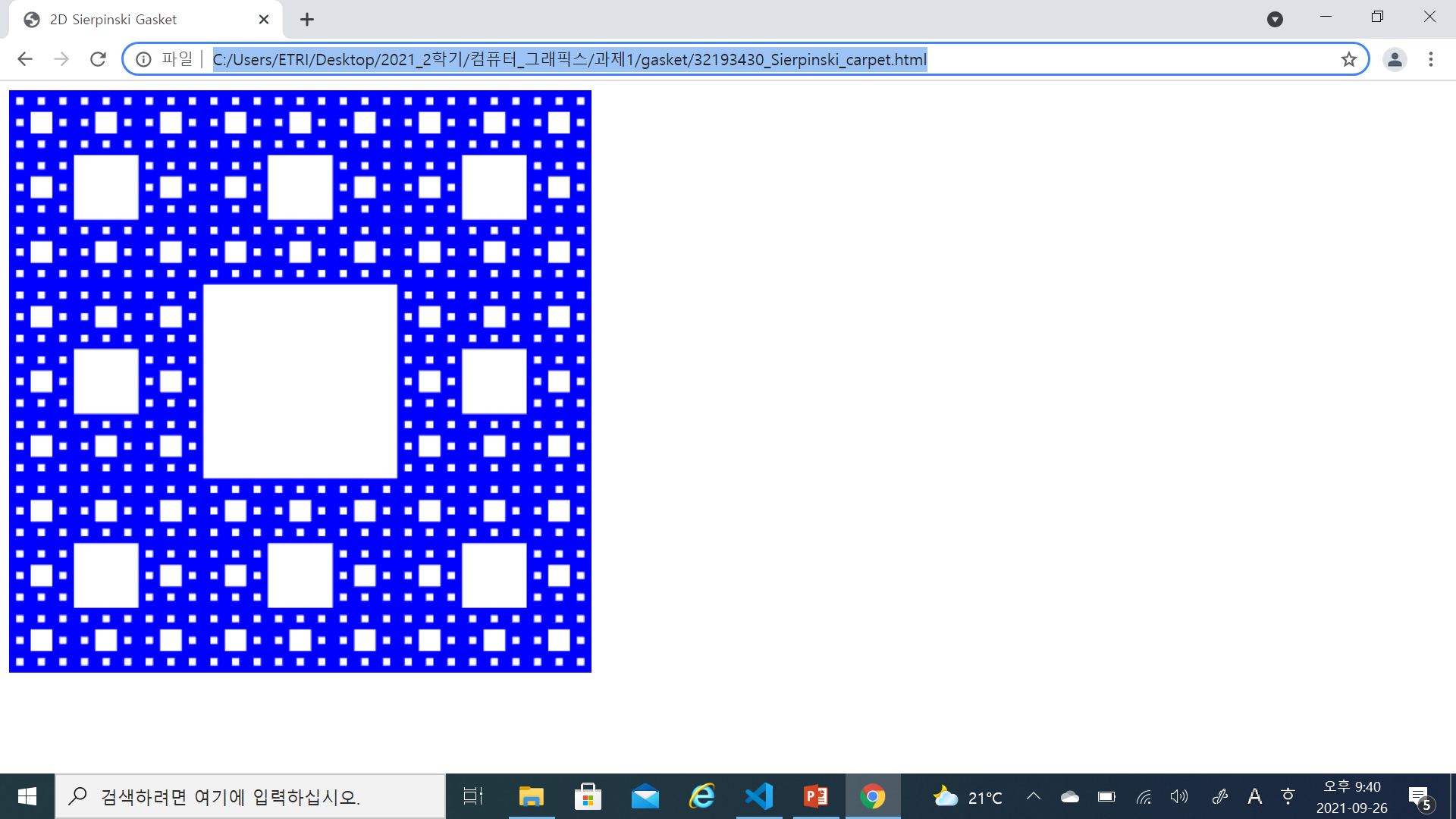
* **divideTriangle 함수 한 번만 사용하였을 경우**

****

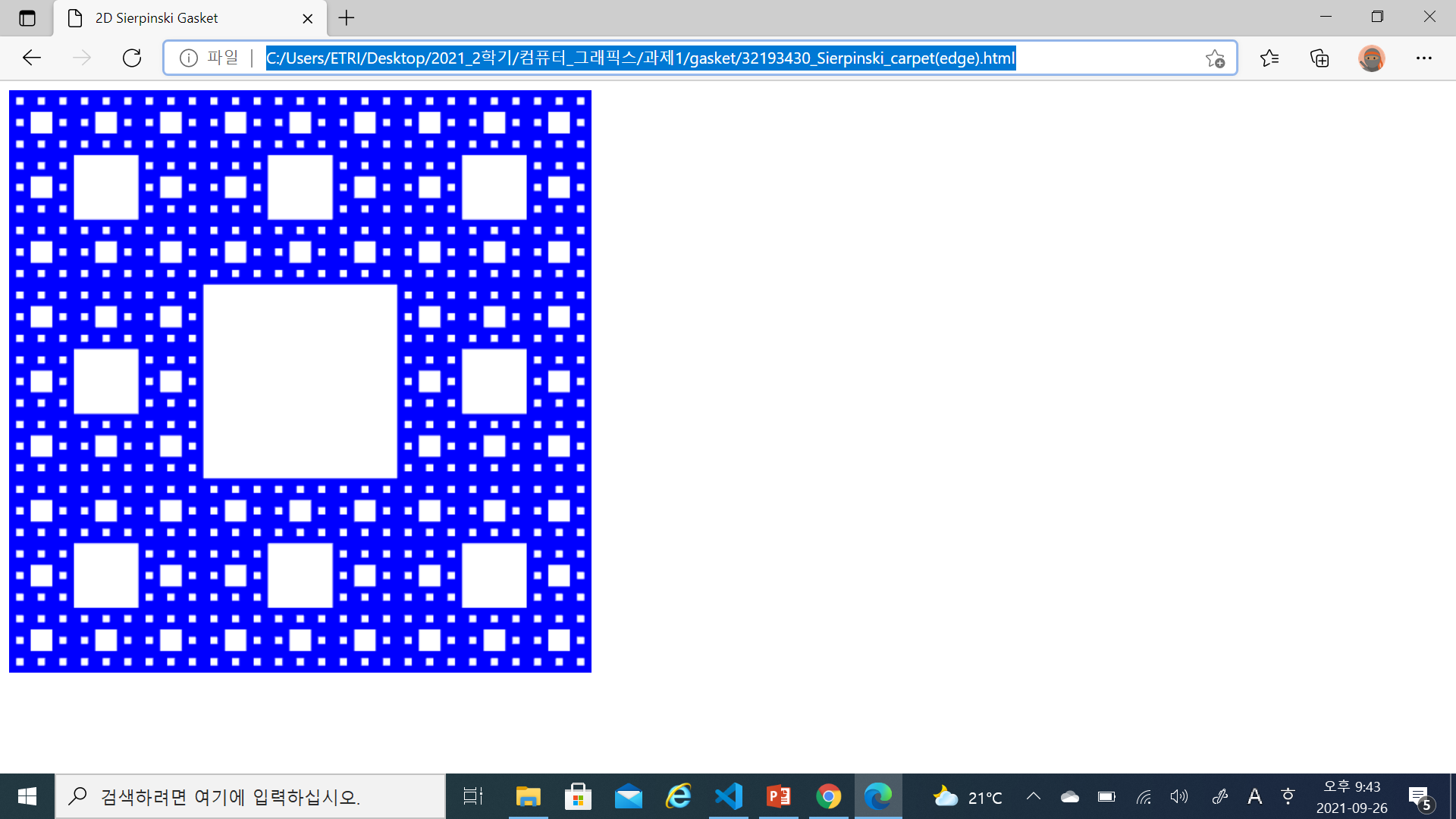
* **divideTriangle 함수 두 번 사용하였을 경우**

****

1. **브라우저 테스트 결과(Chrome, Edge)**
2. **Chrome**

****

1. **Edge**

****

**참고문헌**

Edward Angel, <Interactive Computer Graphics: A Top-down Approach with WebGL 7th edition>, Pearson Education, 2015

<https://en.wikipedia.org/wiki/Sierpi%C5%84ski_carpet>

<https://www.cs.unm.edu/~angel/BOOK/INTERACTIVE_COMPUTER_GRAPHICS/SEVENTH_EDITION/CODE/>