<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Document</title>

<script>

let ctx;

//

let drawables;

let smoke; // 先準備好圖片!

window.addEventListener("load", function(){

ctx = document.getElementById("cvs").getContext("2d");

smoke = new Image();

smoke.src = "smoke.png";

smoke.onload = function(){ // 一開始就準備好

drawables = [];

window.setInterval(refresh, 20); // 每20毫秒執行1次>> 及1秒跑50次

}

})

function refresh(){

drawables.push(new Smoke(ctx.canvas.width/2, 430)) // 畫面放在中見篇下方

// 更新資料

for(let i = 0; i < drawables.length; i++){

if(drawables[i].update()){ // 為了回收機制

// 刪除物件

drawables.splice(i, 1);

i--; // 查陣列splice就知道了 避免漏掉

}

}

// 重繪畫面

ctx.clearRect(0, 0, ctx.canvas.width, ctx.canvas.height);

for(let i = 0; i < drawables.length; i++){

drawables[i].draw();

}

}

// Smoke物件的建構式 今天有2個方形就會有2個物件

function Smoke(x, y){

this.x = x;

this.y = y;

this.vx = Math.random() \* 2 - 1; //改由亂數生 -1~1

this.vy = Math.random() \* 2 - 3; // -1~-3 (往上)

this.size = 20;

this.alpha = 1; // 透明度

// 物件會有更新邏輯

this.update = function(){

// 加上速度後:

// this.vy += 0.02; // 這樣就變等加速度拉XD // 煙就先不用加速度了@ 甚至要減速度

this.x += this.vx;

this.y += this.vy;

this.size += 1; // 變大

if(this.alpha > 0.01){ // 沒加會反轉，所以家這行卡死他

this.alpha -= 0.005; // 變透明!

}

// 為了要有回收機制，否則Array會無限增長:

// return true代表要刪除這個物件

return this.x < 0 || this.y < 0;

}

// 物件的繪圖邏輯

this.draw = function(){

ctx.globalAlpha=this.alpha;

//改為畫影像，因為影像要載入時間，所以事先準備!!!! 不可以onload，所有要等地都要事先準備!!! 像是遊戲!!

ctx.drawImage(smoke, this.x-this.size/2, this.y-this.size/2, this.size, this.size); // 尺寸不要固定喔! 因為smoke會擴散

// /2是因為我們一班是用中間思考(smoke圖中央)

}

}

</script>

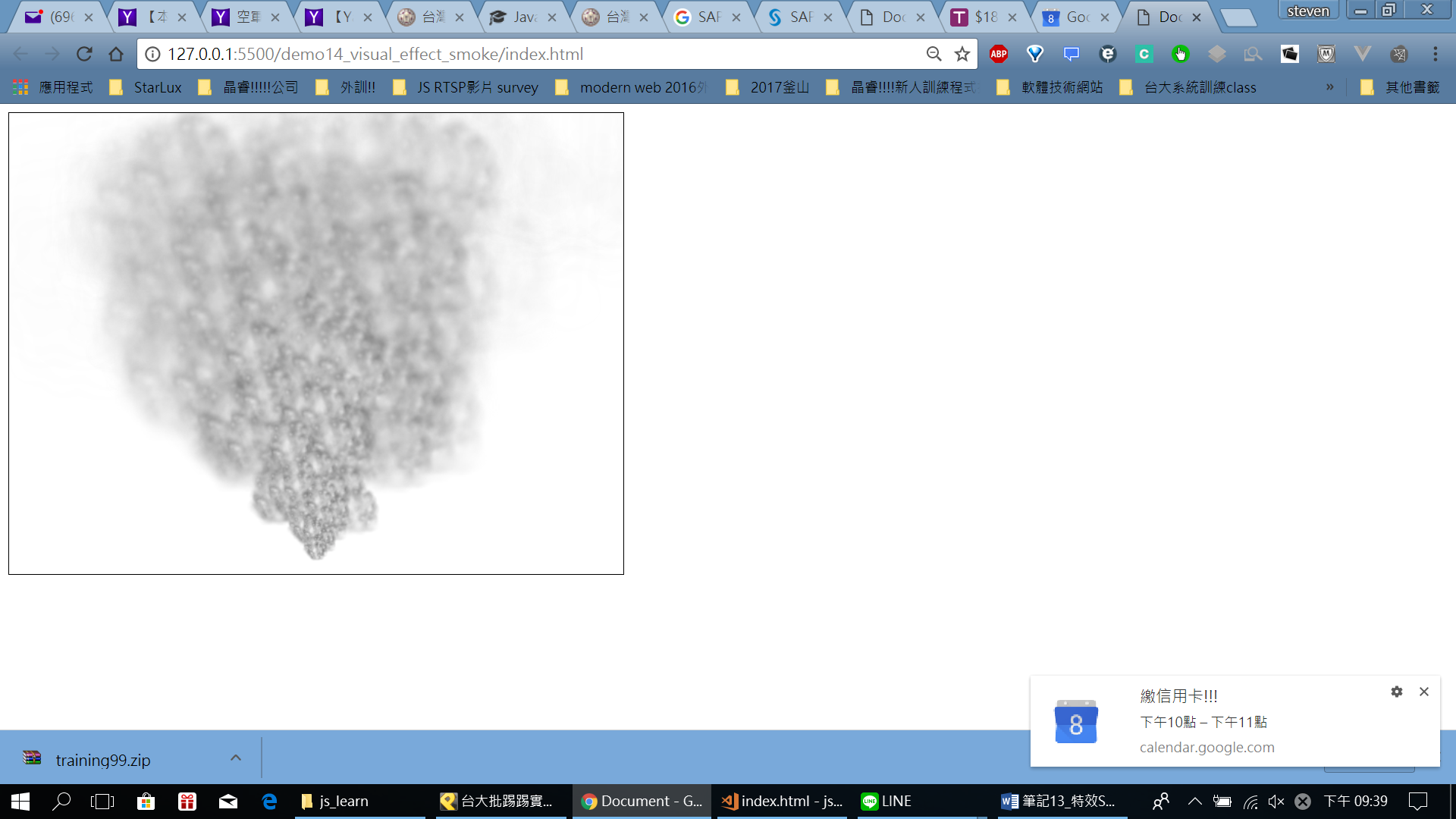
</head>

<body>

<canvas id="cvs" width="600" height="450" style="border:1px solid #000000"></canvas>

</body>

</html>

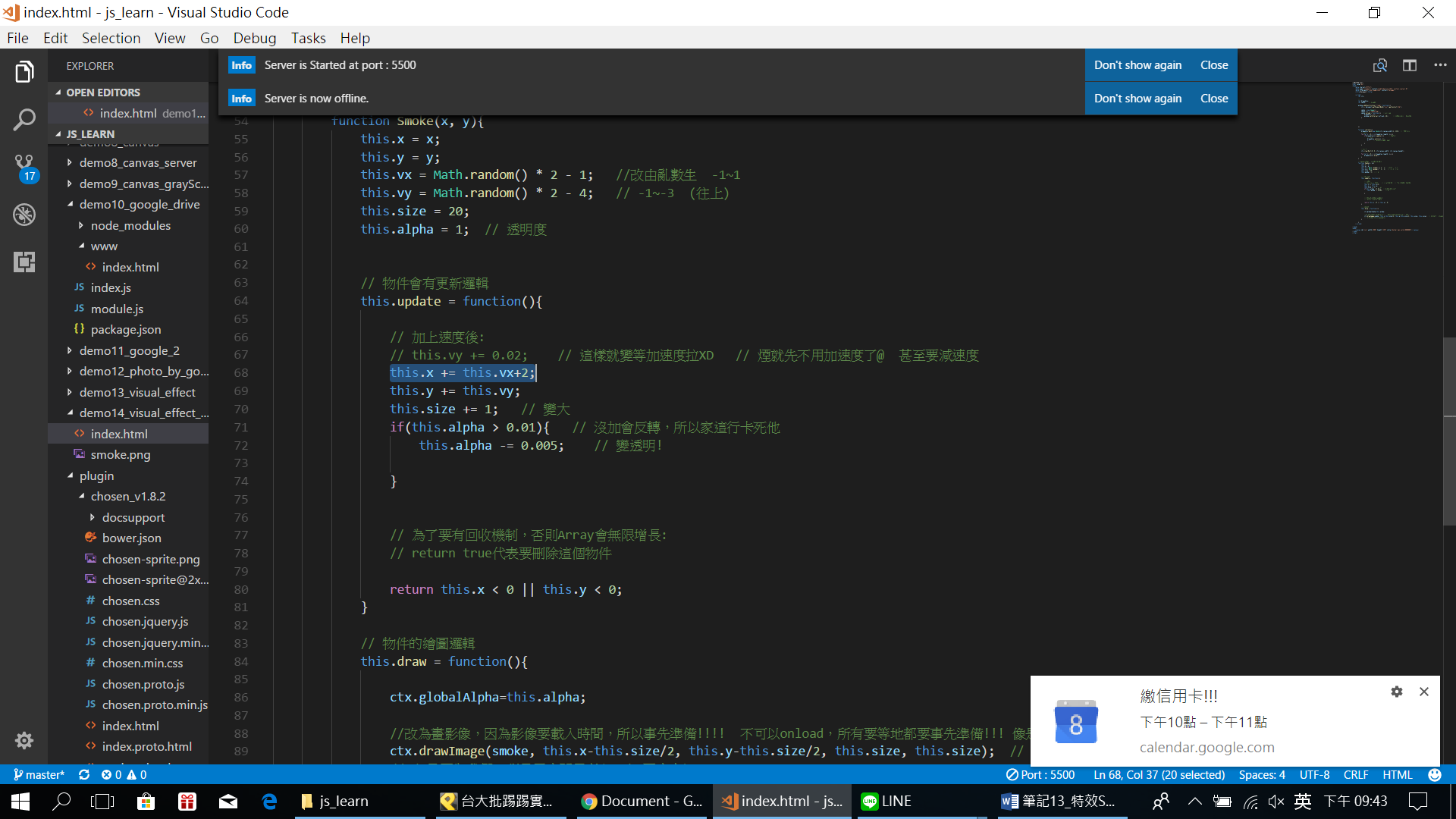


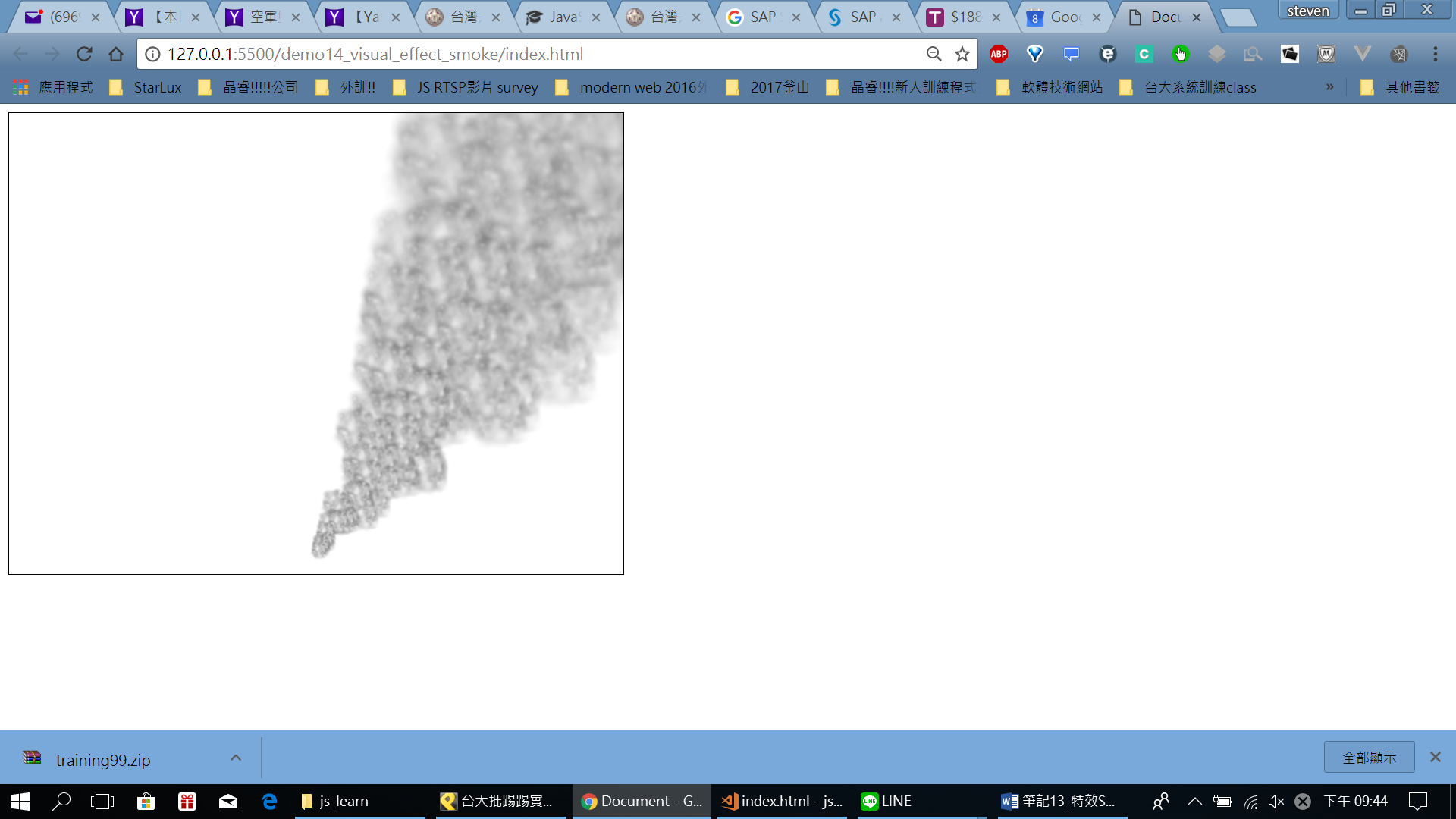
Alpha 再到0的時候可能會反轉到1,

所以卡死他

遊戲的火就是這樣做出來的!!!!!!!!!!!

加風向:





再來改code:

變SmokeSystem:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Document</title>

<script>

let ctx;

//

let drawables;

let smoke; // 先準備好圖片!

window.addEventListener("load", function(){

ctx = document.getElementById("cvs").getContext("2d");

smoke = new Image();

smoke.src = "smoke.png";

smoke.onload = function(){ // 一開始就準備好

drawables = [new SmokeSystem(300, 400)]; // 冒煙點

window.setInterval(refresh, 20); // 每20毫秒執行1次>> 及1秒跑50次

}

})

function refresh(){

// drawables.push(new Smoke(ctx.canvas.width/2, 430)) // 畫面放在中見篇下方 >>> 把他抽出來變新的系統如下!!!

// 更新資料

for(let i = 0; i < drawables.length; i++){

if(drawables[i].update()){ // 為了回收機制

// 刪除物件

drawables.splice(i, 1);

i--; // 查陣列splice就知道了 避免漏掉

}

}

// 重繪畫面

ctx.clearRect(0, 0, ctx.canvas.width, ctx.canvas.height);

for(let i = 0; i < drawables.length; i++){

drawables[i].draw();

}

}

// Smoke System: (包裝用) 把原本Refresh第一列的 drawables.push(new Smoke(ctx.canvas.width/2, 430))包裝

function SmokeSystem(x, y){

// 煙霧系統的冒煙點

this.x = x;

this.y = y;

this.smokes = []; // 煙霧系統底下的所有煙霧分子

this.update = function(){

drawables.push(new Smoke(this.x, this.y)) // 畫面放在中見篇下方

// 更新資料

for(let i = 0; i < this.smokes.length; i++){

if(this.smokes[i].update()){ // 為了回收機制

// 刪除物件

this.smokes.splice(i, 1);

i--; // 查陣列splice就知道了 避免漏掉

}

}

// this.x++;

}

this.draw = function(){

// 更新資料

for(let i = 0; i < this.smokes.length; i++){

this.smokes[i].draw();

}

}

}

// Smoke物件的建構式 今天有2個方形就會有2個物件

function Smoke(x, y){

this.x = x;

this.y = y;

this.vx = Math.random() \* 2 - 1; //改由亂數生 -1~1

this.vy = Math.random() \* 2 - 4; // -1~-3 (往上)

this.size = 20;

this.alpha = 1; // 透明度

// 物件會有更新邏輯

this.update = function(){

// 加上速度後:

// this.vy += 0.02; // 這樣就變等加速度拉XD // 煙就先不用加速度了@ 甚至要減速度

this.x += this.vx; //可以+值便風向!

this.y += this.vy;

this.size += 1; // 變大

if(this.alpha > 0.01){ // 沒加會反轉，所以家這行卡死他

this.alpha -= 0.005; // 變透明!

}

// 為了要有回收機制，否則Array會無限增長:

// return true代表要刪除這個物件

return this.x < 0 || this.y < 0;

}

// 物件的繪圖邏輯

this.draw = function(){

ctx.globalAlpha=this.alpha;

//改為畫影像，因為影像要載入時間，所以事先準備!!!! 不可以onload，所有要等地都要事先準備!!! 像是遊戲!!

ctx.drawImage(smoke, this.x-this.size/2, this.y-this.size/2, this.size, this.size); // 尺寸不要固定喔! 因為smoke會擴散

// /2是因為我們一班是用中間思考(smoke圖中央)

}

}

</script>

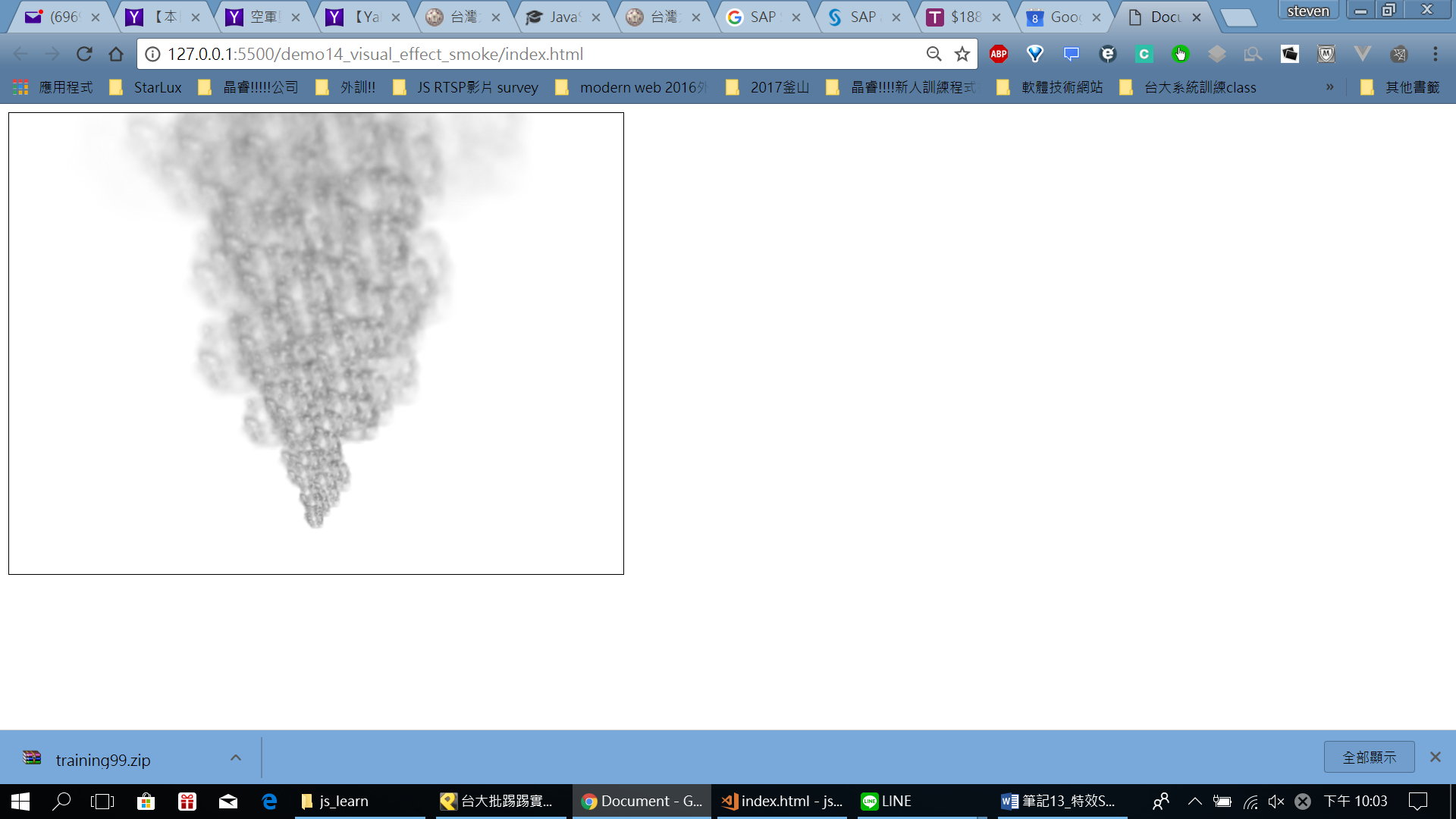
</head>

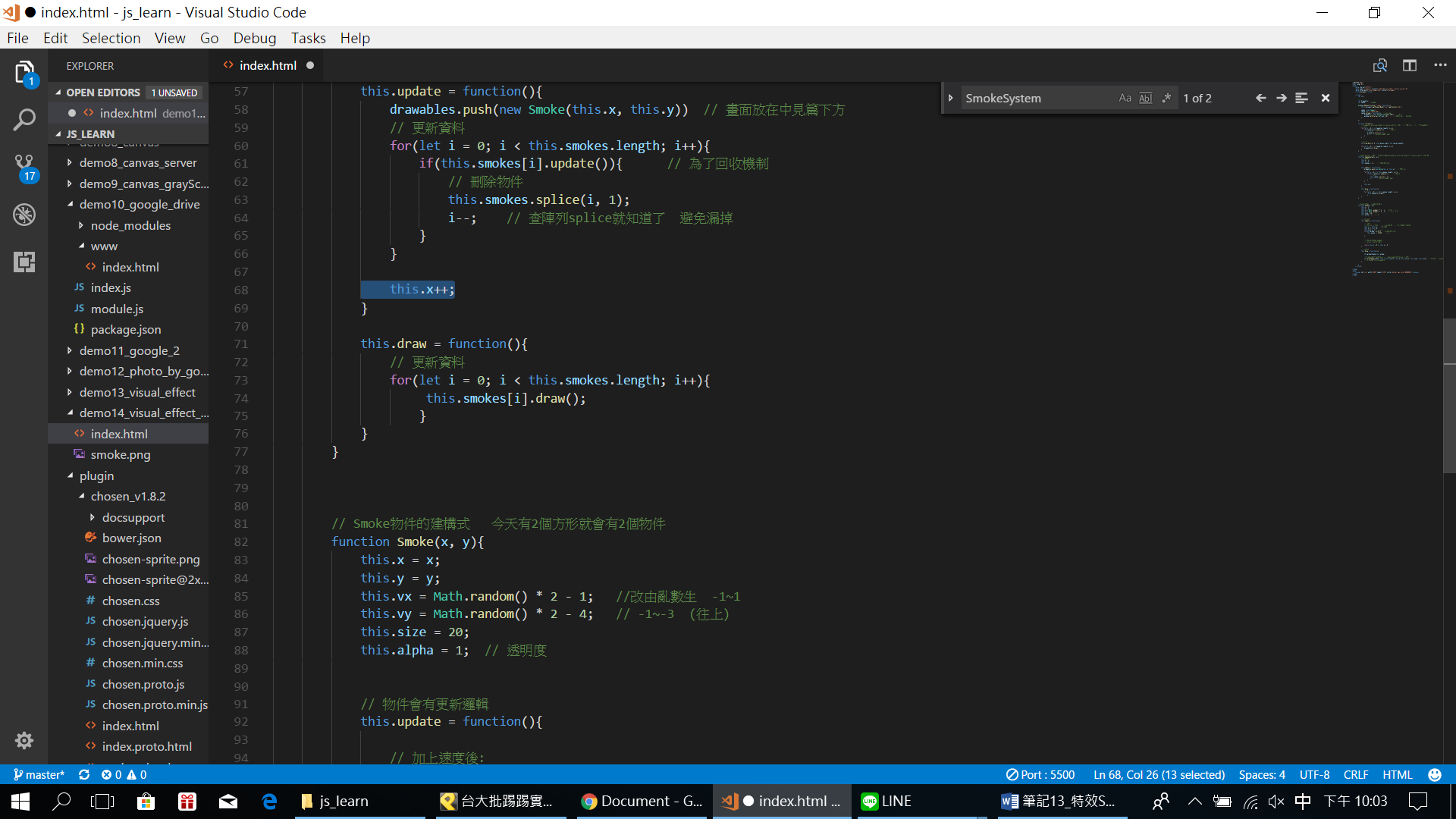
<body>

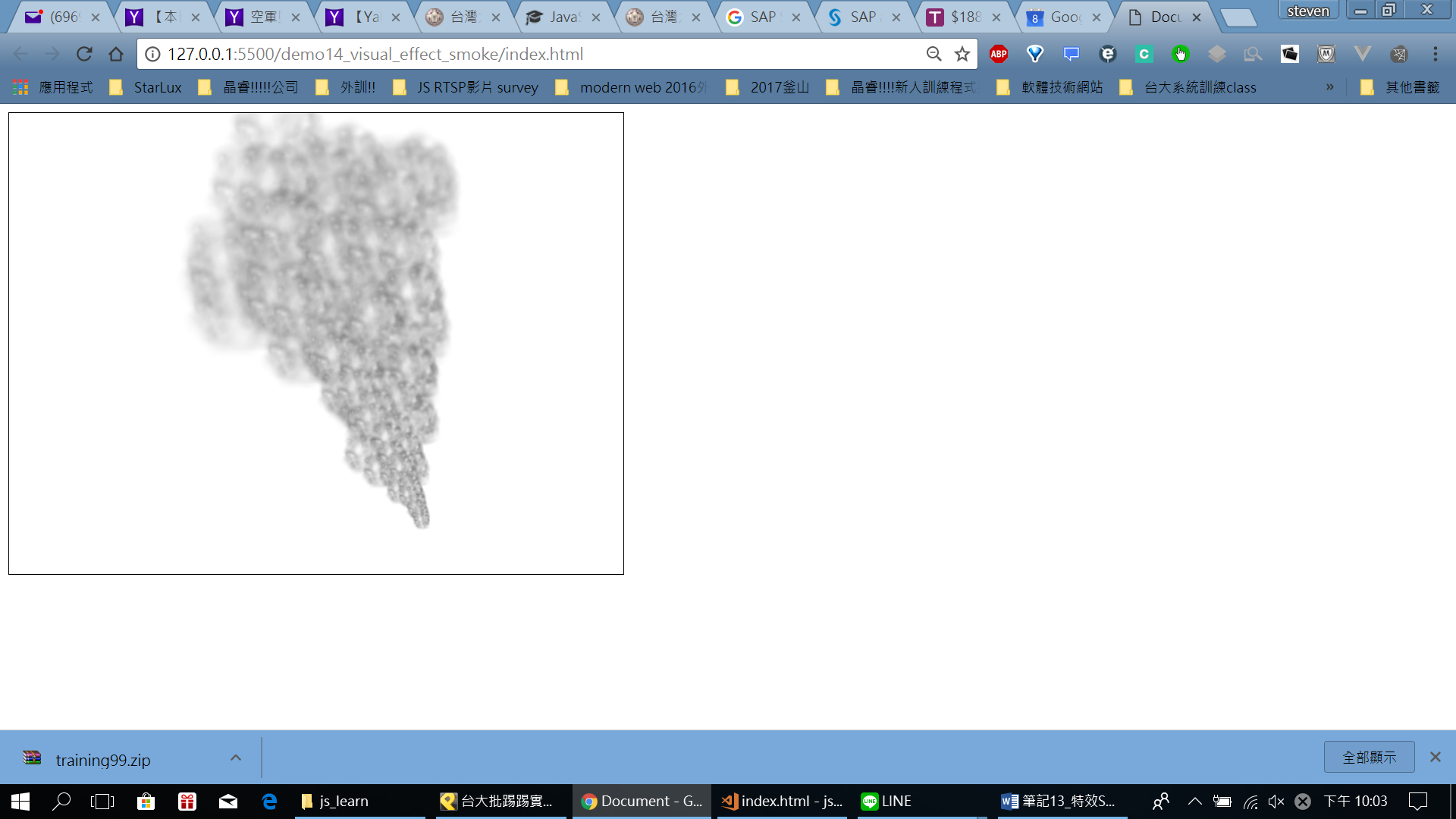
<canvas id="cvs" width="600" height="450" style="border:1px solid #000000"></canvas>

</body>

</html>



也可以在



就會動了!!!