/\*

目標1:存檔連同圖面一起匯入匯出->base64 >OK

目標2:TXT檔案的直接匯入 >OK

目標3:觸控模式下，右鍵選單重新設計->只需要圖片和找泡泡求功能

目標4:手勢操作->放大縮小 ->hammerJS>OK

目標5:整個程式碼重新整理

目標6:匯出存檔壓縮 -> http://ms2.ctjh.ntpc.edu.tw/~luti/105data-compress5.htm

目標7:尋照泡泡求重寫->以尋找BubbleNumber為主

目標8:尋照泡泡求有滑動效果->使用TWEEMJS

目標9:有托曳功能

\*/

canvas2.width = windowWidth;

canvas2.height = windowHeight;

///////////////////////////////////////////////////

image.crossOrigin = "Anonymous";

image.src = "https://i.imgur.com/aS4FwAB.png";

//https://i.imgur.com/kKa6xj6.png

/////////////////////////////////////////////

stage.addChild(pic);

stage.addChild(rightClickMenu);

stage.enableMouseOver(); //重要開啟stage的MouseOver監聽器

// enable touch interactions if supported on the current device:

////////////////////////////////////////////////

var oTable1 = new createjs.DOMElement("style-3");

  oTable1.x = 0;

  oTable1.y = windowHeight-200;

stage.addChild(oTable1);

///////////////////////////////////////

bitmap = new createjs.Bitmap(image);

pic.addChild(bitmap);

///////////////////////////////////////

pic.regX = 278; //定義中心位置(X)

pic.regY = 277.5; //定義中心位置(Y)

pic.x = windowWidth / 2;

pic.y = windowHeight / 2;

pic.scaleX = pic.scaleY = 1.2;

stage.update();

///////////////////////////////////////////

//觸控模式下右鍵選單

var touchMenu = new createjs.Container(); //右鍵選單

stage.addChild(touchMenu);

var touchMenuBackground = new createjs.Shape();

touchMenuBackground.graphics

        .setStrokeStyle(2)

        .beginStroke("#000000")

        .beginFill("DeepSkyBlue")

        .drawRect(0, 0, rightcheckBabblesWidth\*3+rightcheckGap\*4, rightcheckBabblesHeight\*5+rightcheckGap\*6);

touchMenuBackground.alpha = 0.7;

touchMenu.addChild(touchMenuBackground);

touchMenu.x = 100;

touchMenu.y = 100;

var touchinput = new creatNewrightClickButton(rightClickButtonLocation("X",1), rightClickButtonLocation("Y",1), rightcheckBabblesWidth\*3+rightcheckGap\*2, rightcheckBabblesHeight, "#000000", "Lightyellow", findbubble2, "",touchMenu);

var touch9 = new creatNewrightClickButton(rightClickButtonLocation("X",3), rightClickButtonLocation("Y",2), rightcheckBabblesWidth, rightcheckBabblesHeight, "#000000", "Lightyellow", changeTouchNumberTO9, "9",touchMenu);

var touch8 = new creatNewrightClickButton(rightClickButtonLocation("X",2), rightClickButtonLocation("Y",2), rightcheckBabblesWidth, rightcheckBabblesHeight, "#000000", "Lightyellow", changeTouchNumberTO8, "8",touchMenu);

var touch7 = new creatNewrightClickButton(rightClickButtonLocation("X",1), rightClickButtonLocation("Y",2), rightcheckBabblesWidth, rightcheckBabblesHeight, "#000000", "Lightyellow", changeTouchNumberTO7, "7",touchMenu);

var touch6 = new creatNewrightClickButton(rightClickButtonLocation("X",3), rightClickButtonLocation("Y",3), rightcheckBabblesWidth, rightcheckBabblesHeight, "#000000", "Lightyellow", changeTouchNumberTO6, "6",touchMenu);

var touch5 = new creatNewrightClickButton(rightClickButtonLocation("X",2), rightClickButtonLocation("Y",3), rightcheckBabblesWidth, rightcheckBabblesHeight, "#000000", "Lightyellow", changeTouchNumberTO5, "5",touchMenu);

var touch4 = new creatNewrightClickButton(rightClickButtonLocation("X",1), rightClickButtonLocation("Y",3), rightcheckBabblesWidth, rightcheckBabblesHeight, "#000000", "Lightyellow", changeTouchNumberTO4, "4",touchMenu);

var touch3 = new creatNewrightClickButton(rightClickButtonLocation("X",3), rightClickButtonLocation("Y",4), rightcheckBabblesWidth, rightcheckBabblesHeight, "#000000", "Lightyellow", changeTouchNumberTO3, "3",touchMenu);

var touch2 = new creatNewrightClickButton(rightClickButtonLocation("X",2), rightClickButtonLocation("Y",4), rightcheckBabblesWidth, rightcheckBabblesHeight, "#000000", "Lightyellow", changeTouchNumberTO2, "2",touchMenu);

var touch1 = new creatNewrightClickButton(rightClickButtonLocation("X",1), rightClickButtonLocation("Y",4), rightcheckBabblesWidth, rightcheckBabblesHeight, "#000000", "Lightyellow", changeTouchNumberTO1, "1",touchMenu);

var touch0 = new creatNewrightClickButton(rightClickButtonLocation("X",1), rightClickButtonLocation("Y",5), rightcheckBabblesWidth, rightcheckBabblesHeight, "#000000", "Lightyellow", changeTouchNumberTO0, "0",touchMenu);

var touchback = new creatNewrightClickButton(rightClickButtonLocation("X",2), rightClickButtonLocation("Y",5), rightcheckBabblesWidth, rightcheckBabblesHeight, "#000000", "Lightyellow", changeTouchNumberBack, "<-",touchMenu);

var touchDelAll = new creatNewrightClickButton(rightClickButtonLocation("X",3), rightClickButtonLocation("Y",5), rightcheckBabblesWidth, rightcheckBabblesHeight, "#000000", "Lightyellow", touchDelAllButton, "C",touchMenu);

var touchCoverground = new createjs.Shape();

touchCoverground.graphics

        .beginStroke("#000000")

        .beginFill("DeepSkyBlue")

        .drawRect(rightClickButtonLocation("X",1), rightClickButtonLocation("Y",1), rightcheckBabblesWidth\*3+rightcheckGap\*2, rightcheckBabblesHeight\*5+rightcheckGap\*4);

touchMenu.addChild(touchCoverground);

var touchinputSaveData = new creatNewrightClickButton(rightClickButtonLocation("X",1), rightClickButtonLocation("Y",1), rightcheckBabblesWidth\*3+rightcheckGap\*2, rightcheckBabblesHeight\*5+rightcheckGap\*4, "#000000", "Lightyellow", touchDelAllButton, "上傳資料",touchMenu);

var uploadTXTLable3 = new createjs.DOMElement("uploadTXTLable2");

uploadTXTLable3.x = rightClickButtonLocation("X",1);

uploadTXTLable3.y = rightClickButtonLocation("Y",1);

uploadTXTLable3.scaleX = rightcheckBabblesWidth\*3+rightcheckGap\*2/80;

uploadTXTLable3.scaleY =rightcheckBabblesHeight\*5+rightcheckGap\*4/20;

touchMenu.addChild(uploadTXTLable3);

//隱藏選單

touchMenu.visible = false;

stage.update();

///////////////////////////////////////////

var bubblemode = true; //泡泡圖點選模式(默認開啟)

var bubbleShowMode = true;

var bubbleX = []; //儲存滑鼠X座標之陣列

var bubbleY = []; //儲存滑鼠X座標之陣列

var bubbleID = []; //儲存泡泡圖數字座標之陣列

var bubblenumber = []; //儲存泡泡圖數字座標之陣列

var bubbleStroke = []; //儲存泡泡圖線的資料

var bubbleFill = []; //儲存泡泡圖顏色填滿的資料

var bubblesize = []; //儲存泡泡圖大小

var tempID = 1; //暫存現在按排序到第幾個數字

var tempBubbleNumber = 1; //暫存現在存到第幾個數字

var bubbleTempFill = "#ffffff"; //暫存現在的顏色

var bubbleTempStroke = "#000000"; //暫存現在描邊的顏色

var bubbleTempStrokeStyle = 3; //暫存現在描邊的寬度

var bubbleTempSize = 50; //暫存現在泡泡圖大小

var bubbleTextTempSize = 50; //暫存文字大小

var bubbleTextTempColor = "#080808"; //暫存文字顏色

var bubbleTextData = [];

var bubbleTextColorData = [];

var circleAtMouse = new createjs.Shape();

circleAtMouse.graphics

        .setStrokeStyle(5)

        .beginStroke("#fe2712")

        .beginFill("DeepSkyBlue")

        .drawCircle(0, 0, bubbleTempSize);

circleAtMouse.x = 100;

circleAtMouse.y = 100;

var textAtMouse = new createjs.Text(tempBubbleNumber, "50px Arial", bubbleTextTempColor);

textAtMouse.x = 100;

textAtMouse.y = 100;

textAtMouse.textAlign = 'center';

textAtMouse.textBaseline = 'middle';

pic.addChild(circleAtMouse);

pic.addChild(textAtMouse);

createjs.Ticker.on("tick", tick);

createjs.Ticker.setFPS(30);

//創建右鍵選單按鈕

//(X,Y,寬,高,外框顏色,填充顏色,點選時使用的Function,文字內容,要放在哪一個選單)

function creatNewrightClickButton(X, Y, width, height, StrokeColor, FillColor, clickListenerFunction, textContent,inputMenu) {

        this.Shape = new createjs.Shape();

        this.Shape.graphics

                .setStrokeStyle(1)

                .beginStroke(StrokeColor)

                .beginFill(FillColor)

                .drawRect(X, Y, width, height);

    //rightClickMenu.addChild(this.Shape);

        inputMenu.addChild(this.Shape);

        this.Shape.addEventListener('mouseover', handleClick);

        this.Shape.addEventListener('mouseout', handleClick123);

        this.Shape.addEventListener('click', clickListenerFunction);

        this.text = new createjs.Text(textContent, rightcheckBabblesHeight-rightcheckGap+"px Arial", "#080808");

        this.text.x = X+width/2;

        this.text.y = Y+height/2;

        this.text.textAlign = 'center';

        this.text.textBaseline = 'middle';

        inputMenu.addChild(this.text);

}

//計算右鍵選單的位置

function rightClickButtonLocation(rowType,row){

  var outputnumber

  if(rowType == "X"){

    outputnumber = (rightcheckBabblesWidth\*(row-1))+(rightcheckGap\*row);

    return outputnumber

  }

  if(rowType == "Y"){

    outputnumber = (rightcheckBabblesHeight\*(row-1))+(rightcheckGap\*row);

    return outputnumber

  }

}

//在觸控小鍵盤模式下更新輸入數字

function changeTouchNumberTO1(){

  touchinput.text.text = touchinput.text.text+"1";

}

function changeTouchNumberTO2(){

  touchinput.text.text = touchinput.text.text+"2";

}

function changeTouchNumberTO3(){

  touchinput.text.text = touchinput.text.text+"3";

}

function changeTouchNumberTO4(){

  touchinput.text.text = touchinput.text.text+"4";

}

function changeTouchNumberTO5(){

  touchinput.text.text = touchinput.text.text+"5";

}

function changeTouchNumberTO6(){

  touchinput.text.text = touchinput.text.text+"6";

}

function changeTouchNumberTO7(){

  touchinput.text.text = touchinput.text.text+"7";

}

function changeTouchNumberTO8(){

  touchinput.text.text = touchinput.text.text+"8";

}

function changeTouchNumberTO9(){

  touchinput.text.text = touchinput.text.text+"9";

}

function changeTouchNumberTO0(){

  touchinput.text.text = touchinput.text.text+"0";

}

function changeTouchNumberBack(){

  var s

  s =  touchinput.text.text;

  s = s.substring(0,s.length-1);

  touchinput.text.text = s;

}

function touchDelAllButton(){

  touchinput.text.text = "";

}

//創建圓的function

function creatCircle(inpi) {

        //滑鼠座標轉圖面座標

        childCh = pic.globalToLocal(stage.mouseX, stage.mouseY);

        bubbleX[inpi] = childCh.x; //儲存滑鼠X座標之陣列

        bubbleY[inpi] = childCh.y; //儲存滑鼠X座標之陣列

        bubbleID[inpi] = tempID; //儲存泡泡圖的ID

        bubblenumber[inpi] = tempBubbleNumber; //儲存泡泡圖數字座標之陣列

        bubbleStroke[inpi] = bubbleTempStroke; //儲存泡泡圖線顏色的資料

        bubbleFill[inpi] = bubbleTempFill; //儲存泡泡圖顏色填滿的資料

        bubblesize[inpi] = bubbleTempSize; //儲存泡泡圖大小

        bubbleTextData[inpi] = bubbleTextTempSize;

        bubbleTextColorData[inpi] = bubbleTextTempColor;

        var shape = new createjs.Shape();

        shape.graphics.setStrokeStyle(2)

                .beginFill(bubbleFill[inpi])

                .beginStroke(bubbleStroke[inpi]);

        shape.graphics.drawCircle(bubbleX[inpi], bubbleY[inpi], bubblesize[inpi]);

        //shape.x = shapeX[inpi];

        //shape.y = shapeY[inpi];

        ////////////////////////////////////////////////////////////

        var text = new createjs.Text(tempBubbleNumber, bubbleTextData[inpi].toString() + "px Arial", bubbleTextColorData[inpi]);

        text.x = bubbleX[inpi];

        text.y = bubbleY[inpi];

        text.textAlign = 'center';

        text.textBaseline = 'middle';

        /////////////////////////////////////////////////////////

        pic.addChild(shape);

        pic.addChild(text);

}

/////////////////////////////////////////////

//匯出存檔

function outputSaveData() {

        var saveData = "";

        var textFile = null;

        for (var j = 1; j < bubbleID.length; j++) {

                console.log("ID : " + bubbleID[j]);

                console.log("number : " + bubblenumber[j]);

                saveData = saveData +

                        "#bnID" + "#Fnb" + j + "#XFnb" + bubbleID[j] + "#XbnID" + "#Bnb" + j + "#XBnb" +

                        "#bn" + "#Fnb" + j + "#XFnb" + bubblenumber[j] + "#Xbn" + "#Bnb" + j + "#XBnb" +

                        "#bx" + "#Fnb" + j + "#XFnb" + parseInt(bubbleX[j]).toFixed(2) + "#XBx" + "#Bnb" + j + "#XBnb" +

                        "#by" + "#Fnb" + j + "#XFnb" + parseInt(bubbleY[j]).toFixed(2) + "#bxy" + "#Bnb" + j + "#XBnb" +

                        "#bst" + "#Fnb" + j + "#XFnb" + bubbleStroke[j] + "#Xbst" + "#Bnb" + j + "#XBnb" +

                        "#bfc" + "#Fnb" + j + "#XFnb" + bubbleFill[j] + "#Xbfc" + "#Bnb" + j + "#XBnb" +

                        "#bz" + "#Fnb" + j + "#XFnb" + bubblesize[j] + "#Xbz" + "#Bnb" + j + "#XBnb" +

                        "#bt" + "#Fnb" + j + "#XFnb" + bubbleTextData[j] + "#Xbt" + "#Bnb" + j + "#XBnb" +

                        "#btc" + "#Fnb" + j + "#XFnb" + bubbleTextColorData[j] + "#Xbtc" + "#Bnb" + j + "#XBnb";

        }

        saveData = "#all" + bubbleID.length.toString() + "#Xall" + saveData;

    //saveContainToBase64()

     saveData = saveData + "#base64#" + saveContainToBase64() + "#Xbase64#";

        /////////////////////////////////

        //匯出TXT，參考https://stackoverflow.com/questions/21012580/is-it-possible-to-write-data-to-file-using-only-javascript

        var link = document.createElement('a');

        link.setAttribute('download', 'info.txt');

        link.href = makeTextFile(saveData);

        document.body.appendChild(link);

        // wait for the link to be added to the document

        window.requestAnimationFrame(function() {

                var event = new MouseEvent('click');

                link.dispatchEvent(event);

                document.body.removeChild(link);

        });

        ////////////////////////////////

        //console.log(saveData);

}

//匯出TXT，參考https://stackoverflow.com/questions/21012580/is-it-possible-to-write-data-to-file-using-only-javascript

function makeTextFile(text) {

        var textFile = null;

        var data = new Blob([text], {

                type: 'text/plain'

        });

        // If we are replacing a previously generated file we need to

        // manually revoke the object URL to avoid memory leaks.

        if (textFile !== null) {

                window.URL.revokeObjectURL(textFile);

        }

        textFile = window.URL.createObjectURL(data);

        return textFile;

}

/////////////////////////////

//匯入savedata

var inputbubbleID = [],

        inputbubblenumber = [],

        inputbubbleX = [],

        inputbubbleY = [],

        inputbubbleStroke = [],

        inputbubbleFill = [],

        inputbubblesize = [],

        inputbubbleTextdata = [],

        inputbubbleTextColorData = [],

        inputbubbleAllNumber;

var bubbleCircleInPicOrder = [],

        bubbleNumberInPicOrder = [];

//txt匯入教學

//https://www.delftstack.com/zh-tw/howto/javascript/read-text-file-in-javascript/

async function inputSaveData(file) {

        //var textbox = document.getElementById('textbox2');

        //var tempData = textbox.value;

        let tempData = await file.text();

        var a = getSaveDate("#all", "#Xall", tempData);

        inputbubbleAllNumber = a;

        //清空陣列(防誤)

        inputbubbleID.length = 0;

        inputbubblenumber.length = 0;

        inputbubbleX.length = 0;

        inputbubbleY.length = 0;

        inputbubbleStroke.length = 0;

        inputbubbleFill.length = 0;

        inputbubblesize.length = 0;

        inputbubbleTextdata.length = 0;

        inputbubbleTextColorData.length = 0;

        //清空畫布上的泡泡圖(防誤)

        pic.removeAllChildren();

        pic.addChild(bitmap);

        pic.addChild(circleAtMouse);

        pic.addChild(textAtMouse);

        //跟隨的泡泡球變回1(防誤)

        tempBubbleNumber = 1;

        textAtMouse.text = tempBubbleNumber;

    //////////////////////////

    //處理圖片檔BASE64

    var tempbase64 = getSaveDate("#base64#", "#Xbase64#", tempData);

    image.src = tempbase64;

    stage.update();

        /////////////////////

        //資料(savedata)解析並繪出泡泡圖

        for (var j = 1; j < a; j++) {

                inputbubbleID[j] = getSaveDate("#bnID" + "#Fnb" + j + "#XFnb", "#XbnID" + "#Bnb" + j + "#XBnb", tempData);

                inputbubblenumber[j] = getSaveDate("#bn" + "#Fnb" + j + "#XFnb", "#Xbn" + "#Bnb" + j + "#XBnb", tempData);

                inputbubbleX[j] = getSaveDate("#bx" + "#Fnb" + j + "#XFnb", "#XBx" + "#Bnb" + j + "#XBnb", tempData);

                inputbubbleY[j] = getSaveDate("#by" + "#Fnb" + j + "#XFnb", "#bxy" + "#Bnb" + j + "#XBnb", tempData);

                inputbubbleStroke[j] = getSaveDate("#bst" + "#Fnb" + j + "#XFnb", "#Xbst" + "#Bnb" + j + "#XBnb", tempData);

                inputbubbleFill[j] = getSaveDate("#bfc" + "#Fnb" + j + "#XFnb", "#Xbfc" + "#Bnb" + j + "#XBnb", tempData);

                inputbubblesize[j] = getSaveDate("#bz" + "#Fnb" + j + "#XFnb", "#Xbz" + "#Bnb" + j + "#XBnb", tempData);

                inputbubbleTextdata[j] = getSaveDate("#bt" + "#Fnb" + j + "#XFnb", "#Xbt" + "#Bnb" + j + "#XBnb", tempData);

                inputbubbleTextColorData[j] = getSaveDate("#btc" + "#Fnb" + j + "#XFnb", "#Xbtc" + "#Bnb" + j + "#XBnb", tempData);

                //////////////////////////////////////////

                //將通用的繪製資料對齊輸入的資料

                //用於輸入後再增加泡泡圖輸出

                bubbleID[j] = inputbubbleID[j]; //儲存泡泡圖的ID

                bubblenumber[j] = inputbubblenumber[j]; //儲存泡泡圖數字座標之陣列

                bubbleX[j] = inputbubbleX[j]; //儲存滑鼠X座標之陣列

                bubbleY[j] = inputbubbleY[j]; //儲存滑鼠X座標之陣列

                bubbleStroke[j] = inputbubbleStroke[j]; //儲存泡泡圖線顏色的資料

                bubbleFill[j] = inputbubbleFill[j]; //儲存泡泡圖顏色填滿的資料

                bubblesize[j] = inputbubblesize[j]; //儲存泡泡圖大小

                bubbleTextData[j] = inputbubbleTextdata[j];

                bubbleTextColorData[j] = inputbubbleTextColorData[j];

                tempID = tempID + 1; //更新暫存的ID順序

                /////////////////////////////////////////////

                var shape = new createjs.Shape();

                shape.graphics.setStrokeStyle(2)

                        .beginFill(inputbubbleFill[j])

                        .beginStroke(inputbubbleStroke[j]);

                shape.graphics.drawCircle(inputbubbleX[j], inputbubbleY[j], inputbubblesize[j]);

                //shape.x = shapeX[inpi];

                //shape.y = shapeY[inpi];

                ////////////////////////////////////////////////////////////

                var text = new createjs.Text(inputbubblenumber[j], inputbubbleTextdata[j].toString() + "px Arial", inputbubbleTextColorData[j]);

                text.x = inputbubbleX[j];

                text.y = inputbubbleY[j];

                text.textAlign = 'center';

                text.textBaseline = 'middle';

                /////////////////////////////////////////////////////////

                pic.addChild(shape);

                bubbleCircleInPicOrder[j] = pic.numChildren;

                pic.addChild(text);

                bubbleNumberInPicOrder[j] = pic.numChildren;

                ////////////////////////////////////////

                console.log(inputbubblenumber[j] + "," + inputbubbleX[j] + "," + inputbubbleY[j] + "," + inputbubbleStroke[j] + "," + inputbubbleFill[j] + "," + inputbubblesize[j] + "," + inputbubbleTextdata[j] + "," + inputbubbleTextColorData[j]);

        }

        //console.log(tempData);

    //在上傳後隱藏上傳按鈕

    touchCoverground.visible = false;

    touchinputSaveData.Shape.visible = false;

    touchinputSaveData.text.visible = false;

    uploadTXTLable3.visible = false;

}

function getSaveDate(front, back, str) {

        var a = str.search(front);

        var b = str.search(back) + back.length;

        var c = str.slice(a, b);

        var d = c.replace(front, "");

        d = d.replace(back, "");

        return d

}

//inputSaveData();

///////////////////////////////////////////

//相容性寫法，該函式也是網上別人寫的，不過找不到出處了，蠻好的，所有我也沒有必要修改了

//判斷滑鼠滾輪滾動方向

if (testCanvas.addEventListener) //FF,火狐瀏覽器會識別該方法

        testCanvas.addEventListener('DOMMouseScroll', wheel, false);

testCanvas.onmousewheel = testCanvas.onmousewheel = wheel; //W3C

//統一處理滾輪滾動事件

function wheel(event) {

        var delta = 0;

        if (!event) event = testCanvas.event;

        if (event.wheelDelta) { //IE、chrome瀏覽器使用的是wheelDelta，並且值為“正負120”

                delta = event.wheelDelta / 120;

                if (testCanvas.opera) delta = -delta; //因為IE、chrome等向下滾動是負值，FF是正值，為了處理一致性，在此取反處理

        } else if (event.detail) { //FF瀏覽器使用的是detail,其值為“正負3”

                delta = -event.detail / 3;

        }

        if (delta)

                handle(delta);

}

//上下滾動時的具體處理函式

function handle(delta) {

        if (delta < 0) { //向下滾動

                //scalePicture(-0.05, pic);

                scalePicture(countPicSacle(pic, "smaller"), pic);

                //pic.mouseWheel(e => Controls.zoom(controls).worldZoom(e));

        } else { //向上滾動

                //scalePicture(0.05, pic);

                scalePicture(countPicSacle(pic, "bigger"), pic);

                //pic.mouseWheel(e => Controls.zoom(controls).worldZoom(e));

        }

}

//圖片的放大縮小

function scalePicture(scaleNunber, container) {

        //儲存現在縮放量

        let nowX = container.scaleX + scaleNunber;

        let nowY = container.scaleY + scaleNunber;

        if (nowX < 0 || nowY < 0) {

                nowX = container.scaleX - scaleNunber;

                nowY = container.scaleY - scaleNunber;

        } else {

                container.scaleX = nowX;

                container.scaleY = nowY;

                childCh = container.globalToLocal(stage.mouseX, stage.mouseY);

                if (childCh.x > 0 && childCh.x < 2100 && childCh.y > 0 && childCh.y < 2160) {

                        //定位圖片中心至滑鼠位置

                        container.regX = childCh.x;

                        container.regY = childCh.y;

                        //圖片位移至滑鼠位置

                        container.x = stage.mouseX;

                        container.y = stage.mouseY;

                }

        }

        stage.update();

}

//計算圖片放大倍率

function countPicSacle(container, type) {

        /\*

        container.scaleX

        container.scaleY

        canvas2.width

        canvas2.height

        \*/

        //let picwidth = container.scaleX;

        //let picheight = container.scaleY;

        let picwidth = container.scaleX \* image.width;

        let picheight = container.scaleY \* image.height;

        let basicPicSize = 0,

                basicStageSize = 0,

                SacleSize = 0,

                basicImageSize = 0;

        //選定基準數=>以圖片長邊為準

        //if(picwidth > picheight){}

        if (image.width > image.height) {

                basicPicSize = picwidth;

                basicImageSize = image.width;

                basicStageSize = windowWidth;

        } else {

                basicPicSize = picheight;

                basicImageSize = image.height;

                basicStageSize = windowHeight;

        }

        if (type == "bigger") {

                SacleSize = (basicImageSize \* 0.05) / basicImageSize;

                //SacleSize = 0.05;

        }

        if (type == "smaller") {

                SacleSize = -((basicImageSize \* 0.05) / basicImageSize);

                //SacleSize = -0.05;

        }

        if (basicPicSize < basicStageSize) {

                container.regX = image.width / 2;

                container.regY = image.height / 2;

                container.x = windowWidth / 2;

                container.y = windowHeight / 2;

                container.scaleX = basicStageSize / basicImageSize;

                container.scaleY = basicStageSize / basicImageSize;

        }

        //console.log("picwidth"+picwidth);

        //console.log("picheight"+picheight);

        //console.log("basicPicSize"+basicPicSize);

        //console.log("basicStageSize"+basicStageSize);

        //console.log("scaleX"+container.scaleY);

        console.log("image.width : " + image.width);

        console.log("image.height : " + image.height);

        console.log("basicImageSize : " + basicImageSize);

        return SacleSize

}

/////////////////////////////////////

const canvas = document.getElementById('testCanvas');

//禁用右鍵

canvas.oncontextmenu = function(e) {

        return false;

}

//禁止Canvas上滾動

document.addEventListener("mousewheel", (e) => {

        if (e.target === canvas) {

                e.preventDefault()

                        //canvas.zoom(/\*do smth with e\*/)

                console.log('not scrolling')

        } else {

                console.log('scrolling')

        }

}, {

        passive: false

});

/////////////////////////////////////

var initDrag = false;

//滑鼠按下事件

function whichButton(event) {

        var btnNum = event.button;

        if (btnNum == 2) { //2為右鍵

                console.log("您點擊了滑鼠右鍵！");

        } else if (btnNum == 0) { //0為左鍵

                console.log("您點擊了滑鼠左鍵！");

        } else if (btnNum == 1) { //1為中鍵

                console.log("您點擊了滑鼠中鍵！");

                initDrag = true; //拖曳模式開啟

        } else {

                console.log("您點擊了" + btnNum + "號鍵，我不能確定它的名稱。！");

        }

}

function clickup(event) {

        console.log("clickup");

        var btnNum = event.button;

        if (btnNum == 2) { //2為右鍵

                console.log("您放開了滑鼠右鍵！");

                //childCh = pic.globalToLocal(stage.mouseX, stage.mouseY);

        backspaceMode = false;

                rightClickMenu.x = stage.mouseX;

                rightClickMenu.y = stage.mouseY;

        if(stage.mouseX + (rightcheckBabblesWidth\*3+rightcheckGap\*4) > windowWidth){

          rightClickMenu.x = stage.mouseX - (rightcheckBabblesWidth\*3+rightcheckGap\*4);

        }

        if(stage.mouseY + (rightcheckBabblesHeight\*6+rightcheckGap\*7) > windowHeight){

          rightClickMenu.y = stage.mouseY - (rightcheckBabblesHeight\*6+rightcheckGap\*7);

        }

                rightClickMenu.visible = true;

                bubblemode = false;

                stage.update();

        } else if (btnNum == 0) { //0為左鍵

                console.log("您放開了滑鼠左鍵！");

                //console.log(circleAtMouse.Graphics.Fill);

                if (bubblemode == true) {

                        //泡泡球位置對其滑鼠

                        creatCircle(tempID);

                        //outputArray(tempBubbleNumber);

                        //AddTable2(tempBubbleNumber);

                        tempBubbleNumber = tempBubbleNumber + 1;

                        tempID = tempID + 1;

                        textAtMouse.text = tempBubbleNumber;

                        console.log("tempid : " + tempID);

                }

        } else if (btnNum == 1) { //1為中鍵

                console.log("您放開了滑鼠中鍵！");

                initDrag = false; //拖曳模式關閉

        } else {

                console.log("您放開了" + btnNum + "號鍵，我不能確定它的名稱。！");

        }

}

///////////////////////////////////

//螢幕上的鍵盤監聽器

window.addEventListener('keydown', showKey);

function showKey(e) {

        console.log(e.keyCode);

        switch (e.keyCode) {

                //按下backspace鍵

                case 8:

                        if (tempBubbleNumber > 1 && backspaceMode == true) {

                                console.log("backspace");

                                console.log(tempBubbleNumber);

                                var l = pic.numChildren;

                                bubbleChild = pic.getChildAt(l - 1);

                                bubbleChild2 = pic.getChildAt(l - 2);

                                pic.removeChild(bubbleChild);

                                pic.removeChild(bubbleChild2);

                                //text.removeChild(bubbleChildText);

                                tempBubbleNumber = tempBubbleNumber - 1;

                                textAtMouse.text = tempBubbleNumber;

                                tempID = tempID - 1;

                                //去除最後一個元素

                                bubbleID.pop();

                                bubblenumber.pop();

                                bubbleX.pop();

                                bubbleY.pop();

                                bubbleStroke.pop();

                                bubbleFill.pop();

                                bubblesize.pop();

                                bubbleTextData.pop();

                                bubbleTextColorData.pop();

                                stage.update();

                        }

                        break;

                        //按下enter

                case 13:

                        console.log("enter");

                        break;

                        //按下空白建

                case 32:

                        console.log("空白建");

                        bubblemode = true; //開啟泡泡圖模式

                        break;

        }

}

//////////////////////////////////////

//觸控模式

var longTouchMode = false;

var touchTime = 0;

var touchType = "";

if (detectmob() == true) {

        //開啟觸控模式

        createjs.Touch.enable(stage);

        bubbleShowMode = false;

        dragType = "touch";

        document.getElementById("consoleLog").textContent = "開啟觸控模式";

        //參考https://createjs.com/docs/easeljs/classes/DisplayObject.html

        var hammer = Hammer(canvas, {

                prevent\_default: true

        });

        /\*

            hammer.on("touch", startTouches);

            hammer.on("pinch", processPinch);

            hammer.on("drag", processDrag);

            hammer.on("release", clearTouches);

        \*/

        let pan = new Hammer.Pan();

        let pinch = new Hammer.Pinch();

        hammer.add([pan, pinch])

        hammer.get('pinch').set({

                enable: true

        })

        hammer.get('pan').set({

                enable: true

        })

        hammer.on("tap", (e) => {

                document.getElementById("consoleLog").textContent = "tap";

                touchType = "tap";

        })

        hammer.on("press", (e) => {

                document.getElementById("consoleLog").textContent = "press,長按";

                touchType = "press";

                //rightClickMenu.x = 10;

                //rightClickMenu.y = 10;

                //rightClickMenu.visible = true;

        touchMenu.x = 10;

                touchMenu.y = 10;

        touchMenu.visible = true;

                bubblemode = false;

                stage.update();

        })

        hammer.on("panstart", (e) => {

                document.getElementById("consoleLog").textContent = "panstart";

                //longTouchMode = true;

                touchType = "panstart";

                initDrag = false;

        })

        hammer.on("panmove", (e) => {

                document.getElementById("consoleLog").textContent = "panmove,移動";

                touchType = "panmove";

                initDrag = true;

                console.log("panmove");

        })

        hammer.on("panend", (e) => {

                        document.getElementById("consoleLog").textContent = "panend,結束";

                        touchType = "panend";

                        initDrag = false;

                        console.log("panmove");

                })

                /\*

                pinchstart: 多点触控开始

                pinchmove: 多点触控过程

                pinchend: 多点触控结束

                pinchcancel: 多点触控取消

                pinchin: 多点触控时两手指越来越近

                pinchout: 多点触控时两手指越来越远

                \*/

        hammer.on("pinchstart", (e) => {

                document.getElementById("consoleLog").textContent = "pinchstart";

                touchType = "pinchstart";

                //計算中心位置

                var point = pic.globalToLocal(e.center.x, e.center.y);

                pic.regX = point.x;

                pic.regY = point.y;

                pic.x = e.center.x;

                pic.y = e.center.y;

        })

        hammer.on("pinchin", (e) => {

                touchType = "pinchin";

                document.getElementById("consoleLog").textContent = "pinchin,縮小 : " + e.center.x + "," + e.center.y;

                //以中心位置縮小

                scalePicture(countPicSacle(pic, "smaller"), pic);

                /\*

  //定位圖片中心至縮放中心

    pic.regX = e.center.x;

    pic.regY = e.center.y;

  scalePicture(-0.01, pic);

  \*/

        })

        hammer.on("pinchout", (e) => {

                        touchType = "pinchout";

                        document.getElementById("consoleLog").textContent = "pinchout,放大 : " + e.center.x + "," + e.center.y;

                        //以中心位置放大

                        scalePicture(countPicSacle(pic, "bigger"), pic);

                        //scalePicture(0.01, pic);

                        /\*

  //定位圖片中心至縮放中心

    pic.regX = e.center.x;

    pic.regY = e.center.y;

  scalePicture(0.01, pic);

  \*/

                })

                /\*

        stage.on("mousedown", function(evt) {

                longTouchMode = true;

                initDrag = false;

                document.getElementById("consoleLog").textContent = "mousedown";

        });

        stage.on("pressmove", function(evt) {

                initDrag = true;

                touchTime = 0;

                //initDrag = false;

                document.getElementById("consoleLog").textContent = "pressmove";

        });

        stage.on("pressup", function(evt) {

                longTouchMode = false;

                touchTime = 0;

                initDrag = false;

                document.getElementById("consoleLog").textContent = "pressup";

        });

    \*/

}

//偵測使用者的裝置是否為行動裝置

//https://tso1158687.github.io/blog/2019/03/10/detect-mobile-device/

function detectmob() {

        if (navigator.userAgent.match(/Android/i) ||

                navigator.userAgent.match(/webOS/i) ||

                navigator.userAgent.match(/iPhone/i) ||

                navigator.userAgent.match(/iPad/i) ||

                navigator.userAgent.match(/iPod/i) ||

                navigator.userAgent.match(/BlackBerry/i) ||

                navigator.userAgent.match(/Windows Phone/i)

        ) {

                return true;

        } else {

                return false;

        }

}

////////////////////////////////

//右鍵選單的function

//滑鼠在按鈕上時變更顏色(變淡)

function handleClick(event) {

        console.log("evt.target: " + event.target + ", evt.type: " + event.type);

        //筆記:如果要用event調用選取元素要用.targe

        event.target.alpha = 0.5;

        console.log("mouseover");

}

//滑鼠離開按鈕時變更顏色(回復)

function handleClick123(event) {

        event.target.alpha = 1;

        console.log("mouseout");

}

function doNotThing(event) {

        console.log("doNotThing");

}

function rightClickOk(event) {

    backspaceMode = true;//開啟backspace功能

        var paintColor = document.querySelector("#inputColor").value;

        bubbleTempStroke = paintColor;

        console.log("click");

        rightClickMenu.visible = false; //隱藏右鍵選單

        bubblemode = true; //開啟泡泡圖模式

        //如果是觸控模式的話，隱藏泡泡圖

        if (detectmob() == true) {

                bubbleShowMode = false;

        }

}

function bubbleBigger() {

        bubbleTempSize = bubbleTempSize + 3;

        circleAtMouse.scaleX = dimToScale(50, bubbleTempSize);

        circleAtMouse.scaleY = dimToScale(50, bubbleTempSize);

        bubbleTextTempSize = bubbleTextTempSize + 3;

        textAtMouse.scaleX = dimToScale(50, bubbleTempSize);

        textAtMouse.scaleY = dimToScale(50, bubbleTempSize);

        console.log(circleAtMouse.scaleX);

        stage.update(event);

}

function bubbleSmaller() {

        bubbleTempSize = bubbleTempSize - 3;

        circleAtMouse.scaleX = dimToScale(50, bubbleTempSize);

        circleAtMouse.scaleY = dimToScale(50, bubbleTempSize);

        bubbleTextTempSize = bubbleTextTempSize - 3;

        textAtMouse.scaleX = dimToScale(50, bubbleTempSize);

        textAtMouse.scaleY = dimToScale(50, bubbleTempSize);

        stage.update(event);

}

//變更泡泡求數字

function changeNumber() {

        //tempBubbleNumber

        var textTemp = document.getElementById("textbox2");

        var a = Number(textTemp.value);

        console.log(textTemp.value);

        tempBubbleNumber = a;

        textAtMouse.text = tempBubbleNumber;

        stage.update(event);

}

/////////////////////////////////////

function tick(event) {

        //滑鼠座標轉圖面座標

        childCh = pic.globalToLocal(stage.mouseX, stage.mouseY);

        ///////////////////////////////////////////////////////////

        //圖面拖曳效果 -> 滑鼠

        if (initDrag == false && dragType == "mouse") {

                //此方法對齊圖片中心和滑鼠位置

                //定位圖片中心至滑鼠位置

                pic.regX = childCh.x;

                pic.regY = childCh.y;

                //圖片位移至滑鼠位置

                pic.x = stage.mouseX;

                pic.y = stage.mouseY;

        }

        if (initDrag == true && dragType == "mouse") {

                pic.x = stage.mouseX;

                pic.y = stage.mouseY;

                console.log("X:" + pic.x);

                console.log("Y:" + pic.y);

        }

        //圖面拖曳效果 -> 觸控

        if (initDrag == false && dragType == "touch") {

                if (touchType == "panstart" || touchType == "panend") {

                        //在滑動開始即結束時對齊圖片中心和滑鼠位置

                        //定位圖片中心至滑鼠位置

                        pic.regX = childCh.x;

                        pic.regY = childCh.y;

                        //圖片位移至滑鼠位置

                        pic.x = stage.mouseX;

                        pic.y = stage.mouseY;

                }

        }

        if (initDrag == true && dragType == "touch") {

                pic.x = stage.mouseX;

                pic.y = stage.mouseY;

        }

        ///////////////////////////////////////////////////

        //泡泡球位置對齊滑鼠

        circleAtMouse.x = childCh.x;

        circleAtMouse.y = childCh.y;

        textAtMouse.x = circleAtMouse.x;

        textAtMouse.y = circleAtMouse.y;

        //每禎檢查是否為在泡泡圖模式

        if (bubbleShowMode == false) {

                circleAtMouse.visible = false;

                textAtMouse.visible = false;

        }

        ///////////////////////////////////////////////////

        //舞台更新

        stage.update(event);

        /\*

    //console.log(pic.scaleX.toFixed(2));

        //console.log(pic.scaleY.toFixed(2));

        //console.log("X:"+pic.x);

        //console.log("Y:"+pic.y);

        //console.log("regX:"+pic.regX);

        //console.log("regY:"+pic.regY);

        //console.log("childCh.x:"+childCh.x);

        //console.log("childCh.y:"+childCh.y);

        //console.log("滑鼠X:"+stage.mouseX);

        //console.log("滑鼠Y:"+stage.mouseY);

    \*/

        //console.log("滑鼠X:"+stage.mouseX);

        //console.log("滑鼠Y:"+stage.mouseY);

}

///////////////////////////////////////////////////////////////////////////////////////////////

//上傳圖片作為canvas地背景

var form = new FormData(); //通過HTML表單建立FormData物件

var url = '127.0.0.1:8080/'

function selectFile() {

        var files = document.getElementById('ppic').files;

        //console.log(files[0]);

        if (files.length == 0) {

                return;

        }

        var file = files[0];

        //把上傳的圖片顯示出來

        var reader = new FileReader();

        // 將檔案以Data URL形式進行讀入頁面

        //console.log(reader);

        reader.readAsBinaryString(file);

        reader.onload = function(f) {

                        var result = document.getElementById("result");

                        var src = "data:" + file.type + ";base64," + window.btoa(this.result);

                        image.src = src;

                }

                ////////////////////

        image.onload = function() {

                //圖片對齊中間

                pic.x = windowWidth / 2;

                pic.y = windowHeight / 2;

                pic.regX = image.width / 2;

                pic.regY = image.height / 2;

                pic.scaleX = stageWidth / image.width;

                pic.scaleY = pic.scaleX;

                console.log("X:" + pic.x);

                console.log("Y:" + pic.y);

                console.log("regX:" + pic.regX);

                console.log("regY:" + pic.regY);

                stage.update();

                //document.body.appendChild(image);

        }

        ////////////////////

        //console.log('file',file);

        ///////////////////

        form.append('file', file);

        //console.log(form.get('file'));

        if (detectmob() == true) {

                bubbleShowMode = false;

        }

}

/////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////

/////////////////////////////

//圖片匯出功能

//暫存pic並轉存為Base64

function saveContainToBase64() {

        //隱藏滑鼠泡泡球

        circleAtMouse.visible = false;

        textAtMouse.visible = false;

        // Update the stage once to show the loaded image (at its native size)

        stage.update();

        // Cache the bitmap. This is necessary to create the cache-canvas.

        pic.cache(0, 0, image.width, image.height);

        //console.log(image.width + "," + image.height)

        // Note that if you update again, it will show the canvas image as blurred.

        //stage.update();

        // Get the cache-canvas's data url.

        var url = pic.getCacheDataURL();

        return url

}

//將BASE64轉圖片

function outputContainToImage() {

        // Create a new image with the data url, and add it to the body.

        //var img2 = new Image();

        //img2.src = saveContainToBase64();

        //document.body.appendChild(img2);

        var tempfileName = document.getElementById("textbox2");

        if (tempfileName.value == "") {

                tempfileName.value = "未命名";

        } else {

                tempfileName.value = tempfileName.value + " 泡泡圖"

        }

        downloadimage(saveContainToBase64(), tempfileName.value);

}

var saveFile = function(data, filename) {

        //创建一个具有指定的命名空间URI和限定名称的元素。

        var save\_link = document.createElementNS('http://www.w3.org/1999/xhtml', 'a');

        save\_link.href = data;

        save\_link.download = filename;

        var event = document.createEvent('MouseEvents');

        event.initMouseEvent('click', true, false, window, 0, 0, 0, 0, 0, false, false, false, false, 0, null);

        save\_link.dispatchEvent(event);

};

var \_fixType = function(type) {

        type = type.toLowerCase().replace(/jpg/i, 'jpeg');

        var r = type.match(/png|jpeg|bmp|gif/)[0];

        return 'image/' + r;

};

function downloadimage(inputBase64Data, outputFileName) {

        var type = 'png';

        // 返回一個包含JPG圖片的<img>元素

        var img\_png\_src = inputBase64Data; //將畫板儲存為圖片格式的函式

        // 加工image data，替換mime type

        imgData = img\_png\_src.replace(\_fixType(type), 'image/octet-stream');

        // 下載後的問題名

        saveFile(imgData, outputFileName + "." + type);

}

/////////////////////////////////////////////

///////////////////////////////////////////////

///////////////////////////////////

//計算圖片大小

function dimToScale(origDim, desiredDim) {

        return desiredDim / origDim;

}

function scaleToDim(origDim, scale) {

        return scale \* origDim;

}

/////////////////////////////////////////

//變更泡泡求描邊的顏色

function changeButtonColorToRed() {

        bubbleTempStroke = "#ff1571";

        document.querySelector("#inputColor").value = "#ff1571";

        console.log("changeColor");

}

function changeButtonColorToBlue() {

        bubbleTempStroke = "#0084dd";

        document.querySelector("#inputColor").value = "#0084dd";

        console.log("changeColor");

}

function changeButtonColorToGreen() {

        bubbleTempStroke = "#0aa660";

        document.querySelector("#inputColor").value = "#0aa660";

        console.log("changeColor");

}

/////////////////////////////////////////

/////////////////////////////////////////////////////////////////////////////////

//////////////////////////////////////////

//////////////////////////////////////////////

//尋找泡泡球

function findbubble2() {

        //先遍歷匯入的泡泡求陣列的總數，如果要找的數字等於泡泡球的話，

        //找出預先儲存好泡泡球數字在pic這個容器下的順序(bubbleNumberInPicOrder[i])並儲存在tt下，

        //減1是因為不知道為什麼會多1->日後要查

        //然後再將此泡泡求的陣列順序儲存在tempTextSizeData下，

        //方便inputbubbleTextdata(儲存泡泡球數字大小的變數)調用

        var tempTextSizeData, tempCorrectNumber, tt;

        var textTemp = document.getElementById("textbox2");

    var findNumber;

    if(textTemp.value == ""){

      findNumber = touchinput.text.text;

    }else{

      findNumber = textTemp.value;

    }

        for (var i = 0; i < inputbubblenumber.length; i++) {

                if (inputbubblenumber[i] == findNumber) {

                        tt = bubbleNumberInPicOrder[i] - 1;

                        tempCorrectNumber = inputbubblenumber[i];

                        tempTextSizeData = i;

                }

        }

        if (tempCorrectNumber > 0) {

                var childByPic = pic.getChildAt(tt);

                childByPic.alpha = 0.7;

                childByPic.color = "red";

                childByPic.font = inputbubbleTextdata[tempTextSizeData] + "px Impact";

                var pt = pic.localToGlobal(childByPic.x, childByPic.y);

                pic.regX = childByPic.x;

                pic.regY = childByPic.y;

                pic.x = windowWidth / 2;

                pic.y = windowHeight / 2;

        } else {

                alert("資料輸入有誤");

        }

}