

```

/*----ALL VARS----*/

var star;
var starArr = [];
var backdrop;

var starXV;
var starYV;

var starXVArr = [];
var starYVArr = [];

var lineArr = [];

var sliderArr = [];
var sliderLineArr = [];

var NUM_BALLS = 50;
var LINE_DISTANCE = 100;

let linesNumber = 0;
let distanceMultiplier = 1;
let redValue = 255;
let greenValue = 255;
let blueValue = 255;
let ballColor = new Color(redValue,greenValue,blueValue);

let editArea = 150;
let lineSize = getWidth()/3;
let padding = getWidth()/16;

let mouseX;
let mouseY;
let mouseDownBool = false;
let currentMouseElem;

//start function, compiles all functions to make a working program
function start(){
    makeScene();
    mouseStuff();
    setTimer(moveBalls, 20);
}

//creates a line

```

```

function makeEditArea(){
    let topArea = new Line(0, editArea, getWidth(), editArea);
    topArea.setColor(Color.white);
    topArea.setLineWidth(1);
    add(topArea);
}

```

/*adds a simple black backgroud, and calls on functions to create the balls and the lines and sliders*/

```

function makeScene(){
    backdrop = new Rectangle(getWidth(),getHeight());
    backdrop.setColor(Color.black);
    backdrop.setPosition(0,0);
    add(backdrop);
    createBalls();
    makeEditArea();
    makeSliders();
}

```

/*before the loop starts, it goes through the array storing all the lines and removes them from the screen and then from the line array so it is empty when this begins loops through every ball, and for each ball, loops through each ball again and calculates the distance from each ball and the draws a line from each ball if they are close enough and then using the distance, will change the color and thickness of the line, lighter and thicker being closer and darker and thinner being further, an stores each line in an array, and changes the color of the balls based on values gotten from other sliders, at the end, it adds one to a number that keeps track of how many lines have been made (that part is just for fun)*/

```

function createLines(ballArr){
    ballColor = new Color(redValue,greenValue,blueValue);
    for(var k = 0; k < lineArr.length; k++){
        remove(lineArr[k]);
    }
    while(lineArr.length != 0){
        lineArr.pop();
    }
    for(var i = 0; i < ballArr.length; i++){
        var ball = ballArr[i];
        ball.setColor(ballColor);
        var ballX = ball.getX();
        var ballY = ball.getY();
        for(var j = 0; j < ballArr.length; j++){
            if(i != j){

```

```

var ball2 = ballArr[j];
var ball2X = ball2.getX();
var ball2Y = ball2.getY();
var distance = Math.sqrt((ball2X-ballX)**2+(ball2Y-ballY)**2);
if(distance < LINE_DISTANCE){
    var line = new Line(ballX, ballY, ball2X, ball2Y);
    let allColor = 255 - distance/LINE_DISTANCE*255;
    var lineColor = new Color(allColor, allColor, allColor);
    line.setColor(lineColor);
    lineArr.push(line);
    var widthOfLine = 10/distance;
    if(widthOfLine > 2){
        widthOfLine = 2;
    }
    line.setLineWidth(widthOfLine);
    line.layer = Math.round(100/distance);
    add(line);
    linesNumber++;
}
}
}
}
}
}

```

/*creates balls with random positions and sets X and Y velocity above a certain threshold and stores all of it in different arrays*/

```

function createBalls(){
    for(var i = 0; i < NUM_BALLS; i++){
        star = new Circle(Randomizer.nextFloat(1.25,4));
        star.setPosition(Randomizer.nextInt(star.getRadius()+1,getWidth()-1),
            Randomizer.nextInt(star.getRadius()+editArea+1,getHeight()-star.getRadius()-1));
        star.setColor(ballColor);
        star.layer = 11;
        add(star);
        starArr.push(star);
        starXV = Randomizer.nextFloat(-2,2);
        if(starXV > 0){
            while(starXV < .2){
                starXV = Randomizer.nextFloat(0,2);
            }
        }else{
            while(starXV > -.2){

```

```

        starXV = Randomizer.nextFloat(-2,0);
    }
}
starXVArr.push(starXV);
starYV = Randomizer.nextFloat(-2,2);
if(starYV > 0){
    while(starYV < .2){
        starYV = Randomizer.nextFloat(0,2);
    }
}else{
    while(starYV > -.2){
        starYV = Randomizer.nextFloat(-2,0);
    }
}
starYVArr.push(starYV);
}
}

```

/*moves all balls the amount that their voelocity is, and inverts the veolocity if it goes past the end of the grid*/

```

function moveBalls(){
    for(var i = 0; i < starArr.length; i++){
        var ball = starArr[i];
        var XV = starXVArr[i];
        var YV = starYVArr[i];
        ball.move(XV*distanceMultiplier, YV*distanceMultiplier);
        if(ball.getX() > getWidth() - ball.getRadius() || ball.getX() < 0 + ball.getRadius()){
            starXVArr[i] *= -1;
        }
        if(ball.getY() > getHeight() - ball.getRadius() || ball.getY() < 0 + ball.getRadius() +
editArea){
            starYVArr[i] *= -1;
        }
    }
    createLines(starArr);
}

```

/*calculates the percent of the slider on the line and changes the rgb value or the x and y velocity of the balls*/

```

function sliderHandler(){
    for(var i = 0; i < sliderArr.length; i++){
        if(currentMouseElem == sliderArr[i] && mouseDownBool == true){
            let sliderX = mouseX;
            if(sliderX < sliderLineArr[i].getX()){

```

```

        sliderX = sliderArr[i].getX();
    }else if(sliderX > sliderLineArr[i].getX() + lineSize){
        sliderX = sliderLineArr[i].getX() + lineSize;
    }
    sliderArr[i].setPosition(sliderX,sliderArr[i].getY());
    if(i == 0){
        let sliderPercent = (sliderX-sliderLineArr[i].getX())/lineSize;
    }else if(i == 1){
        let sliderPercent = (sliderX-sliderLineArr[i].getX())/lineSize;
        redValue = sliderPercent*255;
    }else if(i == 2){
        let sliderPercent = (sliderX-sliderLineArr[i].getX())/lineSize;
        greenValue = sliderPercent*255;
    }else if(i == 3){
        let sliderPercent = (sliderX-sliderLineArr[i].getX())/lineSize;
        blueValue = sliderPercent*255;
    }
    }
    }
}

```

/*events like mouse clicks and mouse moves can only be used once, so i decided for simplicity to use them all in the same function so it can be accessed all at once, and its much harder to lose*/

```

function mouseStuff(){
    mouseMoveMethod(mouseTracker);
    mouseDownMethod(mouseDown);
    mouseUpMethod(mouseUp);
}

```

/*again, events can only be used once each, so all of the mouse down things are stored in here, but luckily, there arent many of them, it simply gets the shape at the mouse coords, and sets a bool to true*/

```

function mouseDown(e){
    currentMouseElem = getElementAt(mouseX,mouseY);
    mouseDownBool = true;
}

```

//sets the last bool to false

```

function mouseUp(e){
    mouseDownBool = false;
}

```

/*when the mouse moves, it gets the mouse coords and sets variables to be used anywhere in the code and calls on the slider */

```

function mouseTracker(e){

```

```

    mouseX = e.getX();
    mouseY = e.getY();
    sliderHandler();
}

```

/*Creates all the sliders, the lines that they slide on, and the text for each slider and stores them in arrays in order so they can all be accessed easily*/

```

function makeSliders(){
    let sliderText = new Text("Ball Speed","10pt Arial");
    sliderText.setPosition(padding,padding);
    sliderText.setColor(Color.white);
    add(sliderText);

    let sliderLine = new Line(padding, padding+sliderText.getHeight(),
    getWidth()/16+lineSize, padding+sliderText.getHeight());
    sliderLine.setColor(Color.white);
    add(sliderLine);
    sliderLineArr.push(sliderLine);

    let sliderOne = new Circle(5);
    sliderOne.setColor(Color.white);
    sliderOne.setPosition(padding+lineSize/4, padding+sliderText.getHeight());
    add(sliderOne);
    sliderArr.push(sliderOne);

    let sliderTextTwo = new Text("Ball Red Value","10pt Arial");
    sliderTextTwo.setPosition(getWidth()-padding-lineSize,padding);
    sliderTextTwo.setColor(Color.white);
    add(sliderTextTwo);

    let sliderLineTwo = new Line(getWidth()-padding, padding+sliderTextTwo.getHeight(),
    getWidth()-padding-lineSize, padding+sliderTextTwo.getHeight());
    sliderLineTwo.setColor(Color.white);
    add(sliderLineTwo);
    sliderLineArr.push(sliderLineTwo);

    let sliderTwo = new Circle(5);
    sliderTwo.setColor(Color.white);
    sliderTwo.setPosition(getWidth()-padding-1, padding+sliderTextTwo.getHeight());
    add(sliderTwo);
    sliderArr.push(sliderTwo);

    let sliderTextThree = new Text("Ball Green Value","10pt Arial");
    sliderTextThree.setPosition(padding,padding*3.4+sliderText.getHeight());

```

```

sliderTextThree.setColor(Color.white);
add(sliderTextThree);

let sliderLineThree = new Line(padding, padding*4+sliderTextThree.getHeight(),
getWidth()/16+lineSize, padding*4+sliderTextThree.getHeight());
sliderLineThree.setColor(Color.white);
add(sliderLineThree);
sliderLineArr.push(sliderLineThree);

let sliderThree = new Circle(5);
sliderThree.setColor(Color.white);
sliderThree.setPosition(padding+lineSize-1, padding*4+sliderTextThree.getHeight());
add(sliderThree);
sliderArr.push(sliderThree);

let sliderTextFour = new Text("Ball Blue Value","10pt Arial");

sliderTextFour.setPosition(getWidth()-padding-lineSize,padding*3.4+sliderTextFour.getHeight());
sliderTextFour.setColor(Color.white);
add(sliderTextFour);

let sliderLineFour = new Line(getWidth()-padding, padding*4+sliderTextFour.getHeight(),
getWidth()-padding-lineSize, padding*4+sliderTextFour.getHeight());
sliderLineFour.setColor(Color.white);
add(sliderLineFour);
sliderLineArr.push(sliderLineFour);

let sliderFour = new Circle(5);
sliderFour.setColor(Color.white);
sliderFour.setPosition(getWidth()-padding, padding*4+sliderTextFour.getHeight());
add(sliderFour);
sliderArr.push(sliderFour);
}

```