

Component C. Personalized Project Reference.**Procedure:**

i.

```
74 function createLines(ballArr){
75     ballColor = new Color(redValue,greenValue,blueValue);
76     for(var k = 0; k < lineArr.length; k++){
77         remove(lineArr[k]);
78     }
79     while(lineArr.length != 0){
80         lineArr.pop();
81     }
82     for(var i = 0; i < ballArr.length; i++){
83         var ball = ballArr[i];
84         ball.setColor(ballColor);
85         var ballX = ball.getX();
86         var ballY = ball.getY();
87         for(var j = 0; j < ballArr.length; j++){
88             if(i != j){
89                 var ball2 = ballArr[j];
90                 var ball2X = ball2.getX();
91                 var ball2Y = ball2.getY();
92                 var distance = Math.sqrt((ball2X-ballX)**2+(ball2Y-ballY)**2);
93                 if(distance < LINE_DISTANCE){
94                     var line = new Line(ballX, ballY, ball2X, ball2Y);
95                     let allColor = 255 - distance/LINE_DISTANCE*255;
96                     var lineColor = new Color(allColor, allColor, allColor);
97                     line.setColor(lineColor);
98                     lineArr.push(line);
99                     var widthOfLine = 10/distance;
100                     if(widthOfLine > 2){
101                         widthOfLine = 2;
102                     }
103                     line.setLineWidth(widthOfLine);
104                     line.layer = Math.round(100/distance);
105                     add(line);
106                     linesNumber++;
107                 }
108             }
109         }
110     }
111 }
```

ii.

```
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);
}
```

List:

i.

```
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);
  }
}
```

ii.

```
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);
}
```