

```

12 void setup() {
13     size(640, 360);
14     catcher = new Catcher(32); // Create the catcher with a radius of 32
15     drops = new Drop[1000];    // Create 1000 spots in the array
16     timer = new Timer(300);    // Create a timer that goes off every 300 milliseconds
17     timer.start();             // Starting the timer
18 }
19
20 void draw() {}
21     background(255);
22
23     // Set catcher location
24     catcher.setLocation(mouseX, mouseY);
25     // Display the catcher
26     catcher.display();
27
28     // Check the timer
29     if (timer.isFinished()) {
30         // Deal with raindrops
31         // Initialize one drop
32         drops[totalDrops] = new Drop();
33         // Increment totalDrops
34         totalDrops ++ ;
35         // If we hit the end of the array
36         if (totalDrops >= drops.length) {
37             totalDrops = 0; // Start over
38         }
39         timer.start();
40     }
41
42     // Move and display all drops
43     for (int i = 0; i < totalDrops; i++ ) {
44         drops[i].move();
45         drops[i].display();
46         if (catcher.intersect(drops[i])) {
47             drops[i].caught();
48         }
49     }
50 }
51
52
53
54
55
56

```

```
1
2
3
4
5
6
7 class Catcher {
8     float r;    // radius
9     color col;  // color
10    float x, y; // location
11
12    Catcher(float tempR) {
13        r = tempR;
14        col = color(50, 10, 10, 150);
15        x = 0;
16        y = 0;
17    }
18
19    void setLocation(float tempX, float tempY) {
20        x = tempX;
21        y = tempY;
22    }
23
24    void display() {
25        stroke(0);
26        fill(col);
27        ellipse(x, y, r*2, r*2);
28    }
29
30    // A function that returns true or false based on
31    // if the catcher intersects a raindrop
32    boolean intersect(Drop d) {
33        // Calculate distance
34        float distance = dist(x, y, d.x, d.y);
35
36        // Compare distance to sum of radii
37        if (distance < r + d.r) {
38            return true;
39        } else {
40            return false;
41        }
42    }
43 }
44
45
46
47
48
49
50
51
52
53
54
55
56
```

```

1 ,
2 ,
3 ,
4 ,
5 ,
6 ,
7 class Drop {
8     float x, y;    // Variables for location of raindrop
9     float speed;   // Speed of raindrop
10    color c;
11    float r;        // Radius of raindrop
12
13    Drop() {
14        r = 8;                // All raindrops are the same size
15        x = random(width);    // Start with a random x location
16        y = -r*4;             // Start a little above the window
17        speed = random(1, 5); // Pick a random speed
18        c = color(50, 100, 150); // Color
19    }
20
21    // Move the raindrop down
22    void move() {
23        // Increment by speed
24        y += speed;
25    }
26
27    // Check if it hits the bottom
28    boolean reachedBottom() {
29        // If we go a little beyond the bottom
30        if (y > height + r*4) {
31            return true;
32        } else {
33            return false;
34        }
35    }
36
37    // Display the raindrop
38    void display() {
39        // Display the drop
40        fill(c);
41        noStroke();
42        for (int i = 2; i < r; i++ ) {
43            ellipse(x, y + i*4, i*2, i*2);
44        }
45    }
46
47    // If the drop is caught
48    void caught() {
49        // Stop it from moving by setting speed equal to zero
50        speed = 0;
51        // Set the location to somewhere way off-screen
52        y = -1000;
53    }
54 }
55
56

```

```
1
2
3
4
5
6
7 class Timer {
8
9     int savedTime; // When Timer started
10    int totalTime; // How long Timer should last
11
12    Timer(int tempTotalTime) {
13        totalTime = tempTotalTime;
14    }
15
16    // Starting the timer
17    void start() {
18        // When the timer starts it stores the current time in milliseconds.
19        savedTime = millis();
20    }
21
22    // The function isFinished() returns true if 5,000 ms have passed.
23    // The work of the timer is farmed out to this method.
24    boolean isFinished() {
25        // Check how much time has passed
26        int passedTime = millis() - savedTime;
27        if (passedTime > totalTime) {
28            return true;
29        } else {
30            return false;
31        }
32    }
33 }
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
```