

Name: \_\_\_\_\_

## Lab 9 Worksheet

### Exercise 1

In order to have 50 raindrops, which of the animation variables in `starter_raindrop` will become array variables?

Answer: \_\_\_\_\_

Draw a box around the following chunks of code below: (to save space, comments have been omitted – see the provided code)

- the initialization of each animation variable that becomes an array variable
- drawing the circle
- updating the animation variables for the circle

```
float x, y;
float diam;
float c;
boolean fill;

void setup () {
  size(600, 600);

  x = random(0, width);
  y = random(0, height);

  diam = random(0, 50);

  c = random(0, 255);
}

void draw () {
  background(255);

  stroke(c);
  if ( fill ) {
    fill(c);
  } else {
    fill(255);
  }
  ellipse(x, y, diam, diam);

  diam = diam+1;
  c = c+3;
  if ( c >= 255 ) {
    c = 0;
    diam = 0;
    x = random(0, width);
    y = random(0, height);
  }
}

void mouseClicked () {
  fill = !fill;
}
```

## Exercise 2

Which of the animation variables in `starter_delayed` will become array variables? Keep in mind that while the timer is set to a specific value (200) with a single circle, the delays should be staggered so that each circle starts at a different time.

Answer: \_\_\_\_\_

Draw a box around the following chunks of code below: (to save space, comments have been omitted – see the provided code)

- the initialization of each animation variable that becomes an array variable
- drawing the circle
- updating the animation variables for the circle

```
float x, y;
int timer;

void setup () {
  size(400, 400);

  x = random(0, width);
  y = random(0, height);

  timer = 200;
}

void draw () {
  background(255);

  fill(255, 0, 0);
  ellipse(x, y, 10, 10);

  timer = timer-1;

  if ( timer <= 0 ) {
    x = x+1;
  }
}
```

### Exercise 3

A two-stage animation involves a decision: whether to do launch stage stuff or exploding stage stuff. Is this an on-the-spot decision or a prior happenings decision?

Answer: \_\_\_\_\_

Explain your answer – what makes it on-the-spot or prior happenings?

Which of the animation variables in `starter_fireworks` will become array variables?

Answer: \_\_\_\_\_

(continued on the next page)

Draw a box around the following chunks of code below: (to save space, comments have been omitted – see the provided code)

- the initialization of each animation variable that becomes an array variable
- drawing the spark (ellipse)
- updating the animation variables for the spark (ellipse)

Note:

- Be careful not to sweep up too much into the array-ifying loops – non-array animation variables can be used inside array loops but should not be updated or set there.
- There are two local non-array variables – angle in draw() and launchx in reset(). When the firework explodes (the transition from the launch stage to the exploding stage), every spark's speed should be updated with a different random value – so the declaration and initialization of angle should be included in the array loop that updates each spark's speed. When the firework is reset (the transition from exploding to launch), every spark should get the same random x coordinate so the whole firework looks like just a single point during the launch. This means that there should be a single value of launchx for all sparks – so the declaration and initialization of launchx should *not* be included in the array loop that (re)sets each spark's position.

```
float x, y;
float xspeed, yspeed;
float transparency;

float timer;
boolean exploding;

void setup () {
  size(500, 500);

  reset();
  timer = random(50, 150);
  exploding = false;
}
```

```
void reset () {
  float launchx = random(width/4,
                          3*width/4);
  x = launchx;
  y = height;

  xspeed = 0;
  yspeed = -6;

  transparency = 255;
}
```

```
void draw () {
  background(0);

  noStroke();
  fill(255, 255, 0, transparency);
  ellipse(x, y, 4, 4);

  x = x+xspeed;
  y = y+yspeed;
  yspeed = yspeed+.04;
  if ( exploding ) {
    transparency = transparency-1;
  }

  timer = timer-1;

  if ( timer <= 0 && !exploding ) {
    exploding = true;
    timer = 200;

    float angle = random(0, 2*PI);
    xspeed =
      xspeed+cos(angle)*random(0, 1);
    yspeed =
      yspeed+sin(angle)*random(0, 1);

  } else if ( timer <= 0 &&
              exploding ) {
    exploding = false;
    timer = random(50, 150);
    reset();
  }
}
```