

P#6 Fireworks

by Simulation

20201127 이창현



Idea

Seoul International Fireworks Festival



SOURCE : <https://news.lgdisplay.com/2015/09/fireworks-2015/>

Process

Three particle class definitions

- An ascending firecracker
- Exploding firecracker (version 1)
- Exploding firecracker (version 2)

In the main function,

declare the dynamic arrays and output the background image.

When the mouse is pressed,

particle2 is created first.

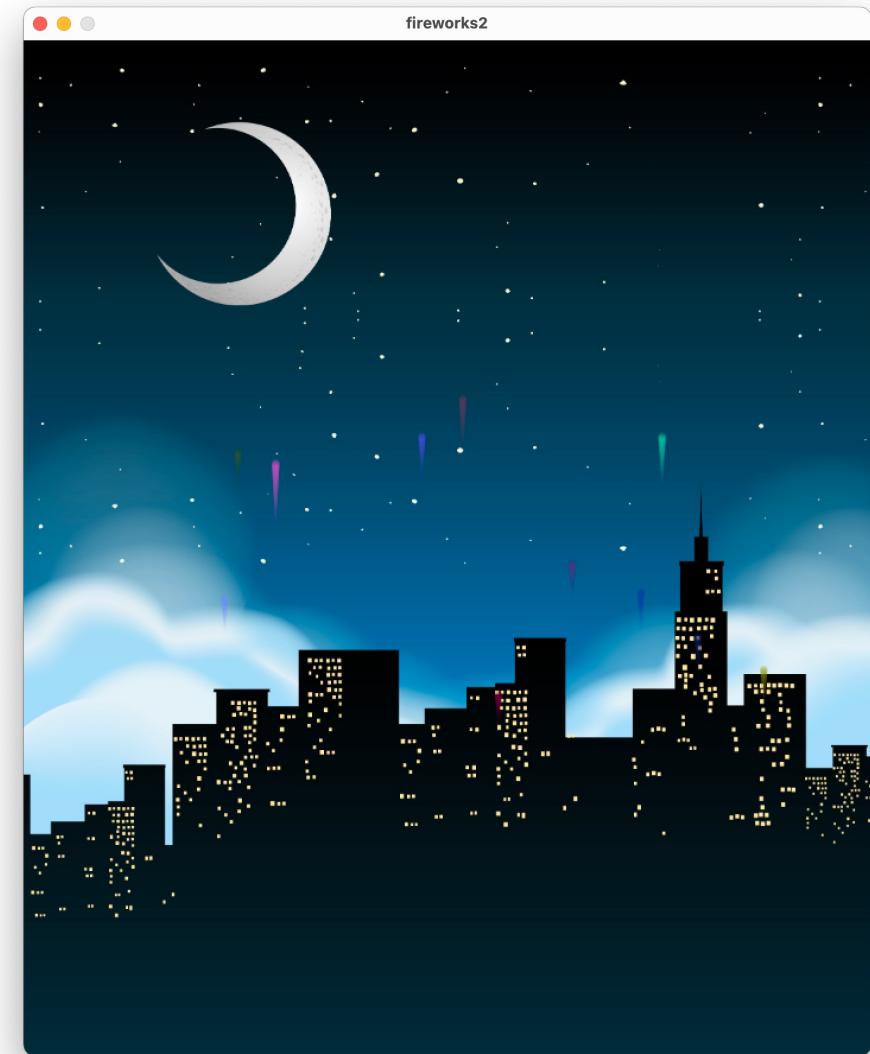
When the y-velocity of particle2 reaches 0, 200 particles of both versions are generated by randomization.

Code - main

```
29 ArrayList<Particle> particles;
30 ArrayList<Particle3> particles3;
31 ArrayList<Particle2> particles2;
32 PImage back;
33
34 void setup() {
35   size(714, 856);
36
37   back = loadImage("night.jpg");
38   back.resize(714, 856);
39
40   particles = new ArrayList<Particle>();
41   particles3 = new ArrayList<Particle3>();
42   particles2 = new ArrayList<Particle2>();
43 }
44
45 void draw() {
46   image(back, 0, 0);
47
48   if (mousePressed) {
49     float col1 = random(255);
50     float col2 = random(255);
51     float col3 = random(255);
52
53     Particle2 t = new Particle2(random(50, width-50), height-100, col1, col2, col3);
54     particles2.add(t); // particles2에 추가
55
56   }
57
58   // particles2 업데이트
59   for (int i = particles2.size() - 1; i >= 0; i--) {
60     Particle2 t = particles2.get(i);
61     t.update();
62     t.display();
63
64     // particles2가 일정 높이에 도달하면 폭발
65     if (t.speedY >= 0) {
66       particles2.remove(i); //우선 올라간 폭죽 제거
67
68       color col = t.col;
69       int x = int(random(100));
70       //int x = 2;
71       //int x = 1;
72       if (x%2 == 1) {
73
74         for (int k = 0; k < 200; k++) {
75           Particle p = new Particle(t.x, t.y, col);
76           particles.add(p);
77         }
78       } else{
79
80         for (int k = 0; k < 200; k++) {
81           Particle3 p = new Particle3(t.x, t.y, col);
82           particles3.add(p);
83         }
84       }
85
86     }
87
88   }
89
90
91   // 폭발한 폭죽 업데이트
92   for (int j = particles.size() - 1; j >= 0; j--) {
93     Particle p = particles.get(j);
94     p.update();
95     p.display();
96
97     // particles가 화면을 벗어나면 제거
98     if (p.y > height) {
99       particles.remove(j);
100    }
101
102   for (int j = particles3.size() - 1; j >= 0; j--) {
103     Particle3 p = particles3.get(j);
104     p.update();
105     p.display();
106
107     // particles가 화면을 벗어나면 제거
108     if (p.y > height) {
109       particles3.remove(j);
110    }
111 }
```

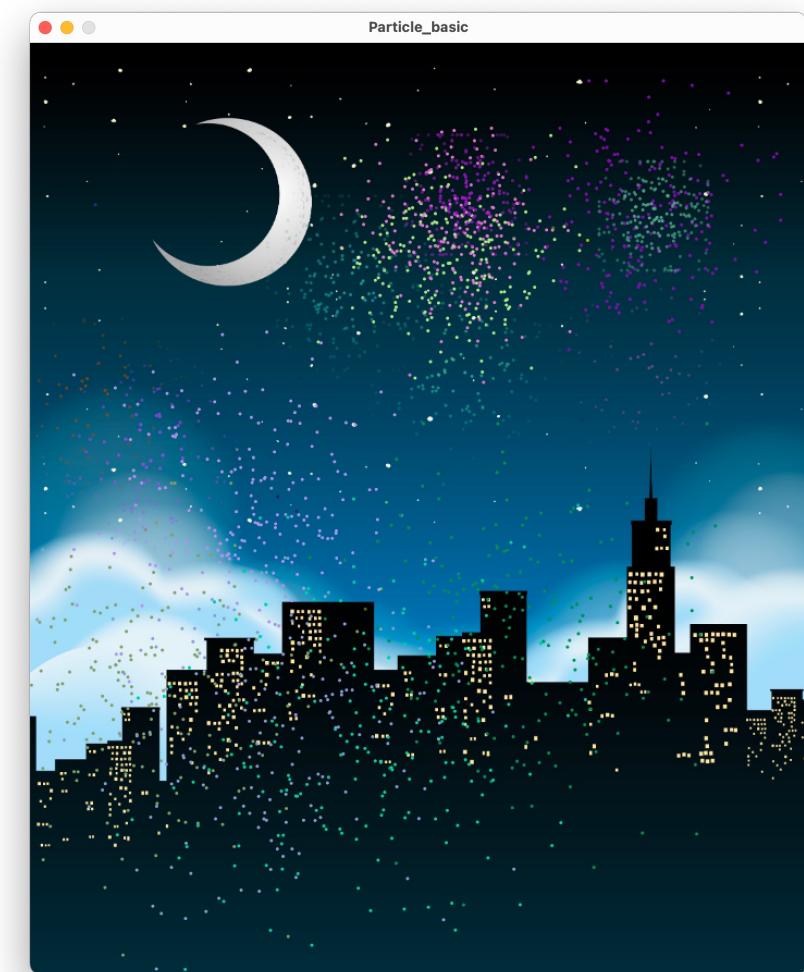
Code - Particle 2(rising firecracker)

```
1 // Particle2 클래스 정의
2 class Particle2 {
3     float x, y; // 위치
4     float speedX, speedY; // 속도
5     float speedY_f;
6     float gravity; // 중력
7     color col; // 색상
8
9     Particle2(float x, float y, float col1, float col2, float col3) {
10         this.x = x;
11         this.y = y;
12         this.speedX = 0;
13         this.speedY = random(-20, -10);
14         this.speedY_f = speedY;
15         this.gravity = 0.3;
16         //this.col = color(random(255), random(255), random(255));
17         this.col = color(col1, col2, col3);
18     }
19
20     void update() {
21         speedY += gravity; // 중력 적용
22         x += speedX;
23         y += speedY;
24     }
25
26     void display() {
27         int movement = int(126*speedY/speedY_f);
28         for( float i = movement; i > 0; i--){
29             float alpha = map(i, movement, 0, 0, 50);
30             fill(col,alpha);
31             noStroke();
32             ellipse(x, y+i, 7-i/(movement/7), 7-i/(movement/7)); // 원 모양의 파티클
33         }
34     }
35 }
```

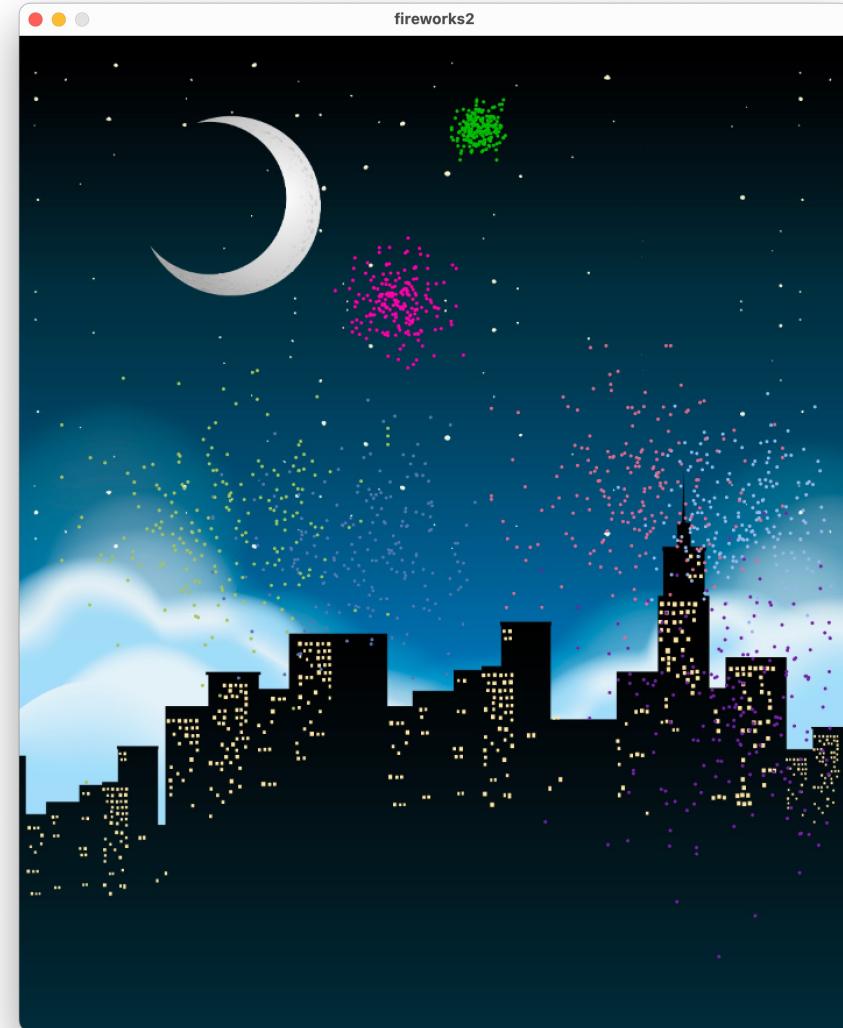
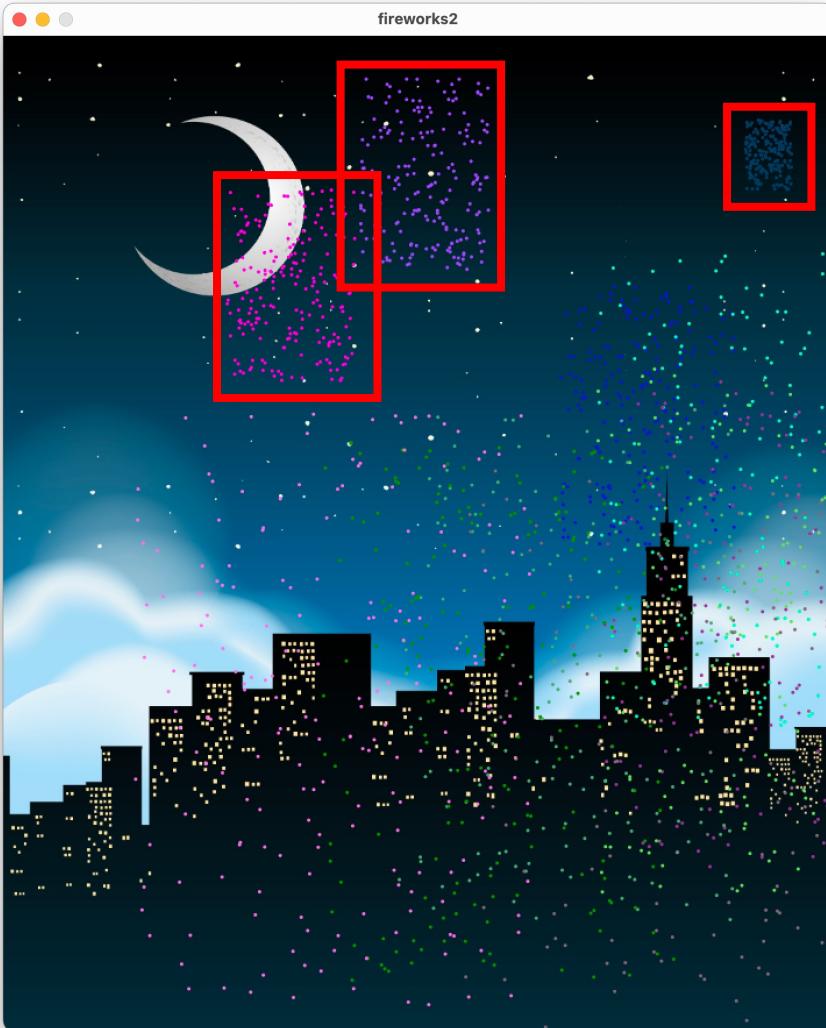


Code - Particle basic (exploding firecracker ver1)

```
1 // Particle 클래스 정의
2 class Particle {
3     float x, y; // 위치
4     float speedX, speedY; // 속도
5     float gravity; // 중력
6     color col; // 색상
7
8     Particle(float x, float y,color col) {
9         this.x = x;
10        this.y = y;
11        this.speedX = constrain(randomGaussian() * 2, -5, 5); // 평균 0, 표준편차 2로 설정
12        this.speedY = constrain(randomGaussian() * 2-5, -10, 4);
13        this.gravity = 0.3;
14        //this.col = color(random(255), random(255), random(255));
15        this.col = col;
16    }
17
18    void update() {
19        speedY += gravity; // 중력 적용
20        x += speedX;
21        y += speedY;
22    }
23
24    void display() {
25        fill(col);
26        noStroke();
27        ellipse(x, y, 3, 3); // 원 모양의 파티클
28    }
29}
```

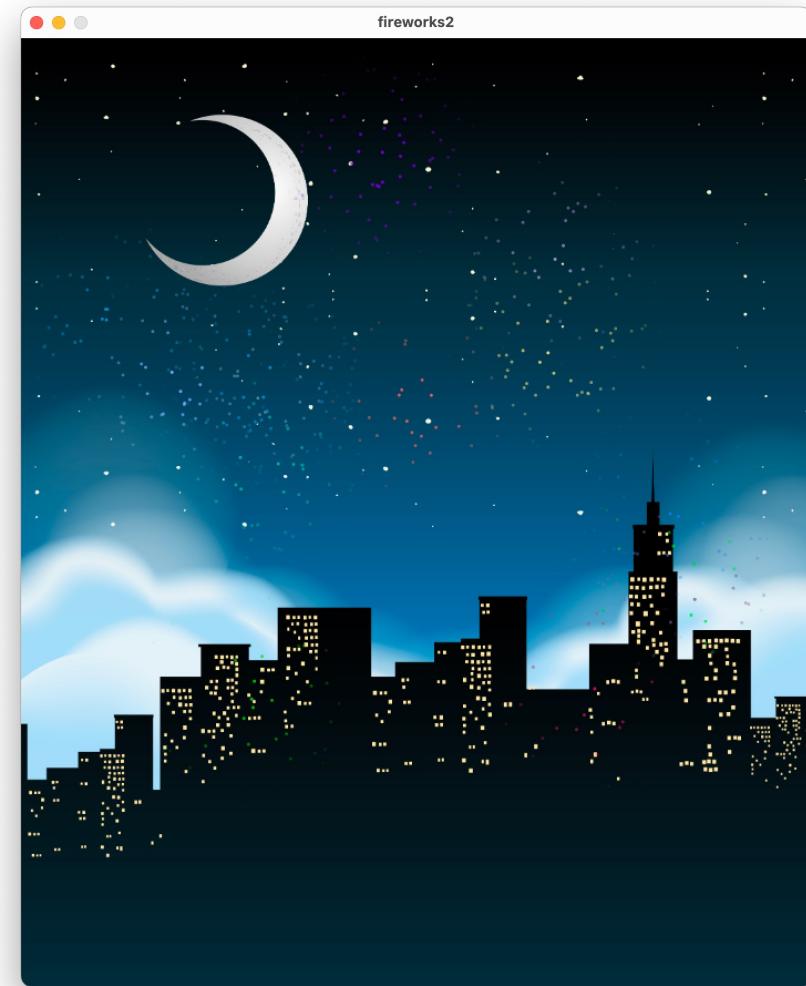


Using randomGaussian()



Code - Particle var1 (exploding firecracker ver1)

```
1 // Particle3 클래스 정의
2 class Particle3 {
3     float x, y; // 위치
4     float x_i,y_i; //초기위치
5     float speedX, speedY; // 속도
6     float gravity; // 중력
7     color col; // 색상
8
9     Particle3(float x, float y,color col) {
10         this.x = x;
11         this.y = y;
12         this.x_i = x;
13         this.y_i = y;
14         this.speedX = constrain(randomGaussian() * 3, -5, 5); // 평균 0, 표준편차 2로 설정
15         this.speedY = constrain(randomGaussian() * 3 ,-5, 5);
16         this.gravity = 0;
17         //this.col = color(random(255), random(255), random(255));
18         this.col = col;
19     }
20
21     void update() {
22         speedY += gravity; // 중력 적용
23         x += speedX;
24         y += speedY;
25     }
26
27     void display() {
28         int movement = int(abs(sqrt((x-x_i)*(x-x_i)+(y-y_i)*(y-y_i)))); // 움직임 계산
29         float alpha = map(movement, 100, 0, 0, 255);
30         fill(col,alpha);
31         noStroke();
32         ellipse(x, y, 3, 3); // 원 모양의 파티클
33     }
34 }
```



Screenshot

