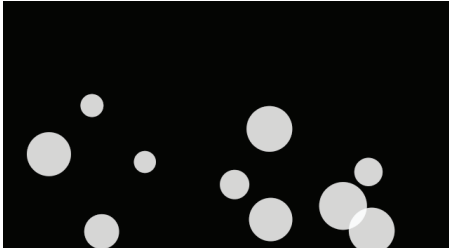


Bouncing Bubbles

CODE



```
/**
 * Bouncy Bubbles
 * based on code from Keith Peters.
 *
 * Multiple-object collision.
 */
```

```
int numBalls = 10;
float spring = 0.05;
float gravity = 0.03;
float friction = -0.9;
Ball[] balls = new Ball[numBalls];

void setup() {
    size(640, 360);
    for (int i = 0; i < numBalls; i++) {
        balls[i] = new Ball(random(width), random(height), random(30,
70), i, balls);
    }
    noStroke();
    fill(255, 204);
}

void draw() {
    background(0);
    for (int i = 0; i < numBalls; i++) {
        balls[i].collide();
        balls[i].move();
        balls[i].display();
    }
}

class Ball {

    float x, y;
    float diameter;
    float vx = 0;
    float vy = 0;
    int id;
    Ball[] others;

    Ball(float xin, float yin, float din, int idin, Ball[] oin) {
        x = xin;
        y = yin;
        diameter = din;
        id = idin;
        others = oin;
    }

    void collide() {
        for (int i = id + 1; i < numBalls; i++) {
            float dx = others[i].x - x;
            float dy = others[i].y - y;
            float distance = sqrt(dx*dx + dy*dy);
            float minDist = others[i].diameter/2 + diameter/2;
            if (distance < minDist) {
                float angle = atan2(dy, dx);
                float targetX = x + cos(angle) * minDist;
                float targetY = y + sin(angle) * minDist;
                float ax = (targetX - others[i].x) * spring;
                float ay = (targetY - others[i].y) * spring;
                vx -= ax;
                vy -= ay;
            }
        }
    }
}
```

```

        others[i].vx += ax;
        others[i].vy += ay;
    }
}

void move() {
    vy += gravity;
    x += vx;
    y += vy;
    if (x + diameter/2 > width) {
        x = width - diameter/2;
        vx *= friction;
    }
    else if (x - diameter/2 < 0) {
        x = diameter/2;
        vx *= friction;
    }
    if (y + diameter/2 > height) {
        y = height - diameter/2;
        vy *= friction;
    }
    else if (y - diameter/2 < 0) {
        y = diameter/2;
        vy *= friction;
    }
}

void display() {
    ellipse(x, y, diameter, diameter);
}
}

```