

在PDF 最下方有源代码可以直接复制并粘贴。

这里是烟花动画链接 <https://huoshi111.github.io/yanhuafireworks/yanhua.html>

不知道如何将代码写入, 可以阅读以下教程

要将代码写入, 可以直接在电脑键盘 **WIN + R** 打开执行窗口, 在执行窗口里搜索写入**notepad**或中文叫做记事本, 在记事本里写入以下代码, 在pdf最下面有显示源代码, 你可以直接复制并粘贴到记事本, 粘贴完后, 记事本左上角有一个按钮叫做文件下面有一个按钮叫做保存为, 点进去, 写下一个名字后面记得写一个**.html** 按保存, 这样就有了。

```
20 <body>
21 <canvas id="canvas"></canvas>
22 <script>
23   window.addEventListener("resize", resizeCanvas, false);
24   window.addEventListener("DOMContentLoaded", onLoad, false);
25   window.addEventListener("click", launchFireworkOnClick, false);
26
27   var canvas, ctx, w, h, particles = [], fireworks = [];
28
29   function onLoad() {
30     canvas = document.getElementById("canvas");
31     ctx = canvas.getContext("2d");
32     resizeCanvas();
33     window.requestAnimationFrame(updateWorld);
34   }
35
36   function resizeCanvas() {
37     if (!canvas) {
38       w = canvas.width = window.innerWidth;
39       h = canvas.height = window.innerHeight;
40     }
41   }
42
43   function launchFireworkOnClick(event) {
44     const clickX = event.clientX;
45     const clickY = event.clientY;
46     fireworks.push(new Firework(clickX, h, clickX, clickY));
47   }
```

```
48
49   function updateWorld() {
50     update();
51     paint();
52     window.requestAnimationFrame(updateWorld);
53   }
54
55   function update() {
56     for (let i = fireworks.length - 1; i >= 0; i--) {
57       if (!fireworks[i].move()) {
58         fireworks.splice(i, 1);
59       }
60     }
61
62     const alive = [];
63     for (let i = 0; i < particles.length; i++) {
64       if (particles[i].move()) {
65         alive.push(particles[i]);
66       }
67     }
68     particles = alive;
69   }
70
71   function paint() {
72     ctx.globalCompositeOperation = "source-over";
73     ctx.clearRect(0, 0, w, h);
74
75     ctx.globalCompositeOperation = "lighter";
76
77     for (let i = 0; i < fireworks.length; i++) {
78       fireworks[i].draw(ctx);
79     }
```

```

80
81     for (let i = 0; i < particles.length; i++) {
82         particles[i].draw(ctx);
83     }
84 }
85
86 function Firework(sx, sy, tx, ty) {
87     this.x = sx;
88     this.y = sy;
89     this.tx = tx;
90     this.ty = ty;
91     this.vx = (tx - sx) / 90;
92     this.vy = (ty - sy) / 40;
93     this.exploded = false;
94     this.color =
95         "rgb(" +
96         (~Math.random() * 200 + 55) +
97         ", " +
98         (~Math.random() * 200 + 55) +
99         ", " +
100         (~Math.random() * 200 + 55) +
101         ")";
102     this.trail = [];
103 }
104
105 Firework.prototype.move = function () {
106     if (!this.exploded) {
107         this.trail.push({ x: this.x, y: this.y, alpha: 1 });

```

```

108
109         if (this.trail.length > 20) {
110             this.trail.shift();
111         }
112
113         this.vx *= 0.98; // 空气阻力
114         this.vy += 0.05; // 模拟重力
115         this.x += this.vx;
116         this.y += this.vy;
117
118         if (Math.abs(this.x - this.tx) < 5 && Math.abs(this.y - this.ty) < 5) {
119             this.explode();
120             this.exploded = true;
121         }
122         return true;
123     }
124     return false;
125 };
126
127 Firework.prototype.draw = function (c) {
128     if (!this.exploded) {
129         c.save();
130
131         // 发射拖尾
132         for (let i = 0; i < this.trail.length; i++) {
133             const t = this.trail[i];
134             c.beginPath();
135             c.arc(t.x, t.y, 2, 0, Math.PI * 2);
136             c.fillStyle = this.color;
137             c.globalAlpha = t.alpha;
138             c.fill();

```

```

138         c.fill();
139         t.alpha -= 0.05;
140     }
141
142     c.restore();
143
144     // 发射点
145     c.beginPath();
146     c.arc(this.x, this.y, 4, 0, Math.PI * 2);
147     c.fillStyle = this.color;
148     c.fill();
149 }
150 };
151
152 Firework.prototype.explode = function () {
153     const nParticles = Math.random() * 150 + 200;
154
155     for (let i = 0; i < nParticles; i++) {
156         const angle = Math.random() * Math.PI * 2;
157         const speed = Math.random() * 5 + 4;
158         const vx = Math.cos(angle) * speed;
159         const vy = Math.sin(angle) * speed;
160         const particle = new Particle(this.x, this.y, vx, vy, this.color);
161         particles.push(particle);
162     }
163 };
164

```

```

164
165 function Particle(x, y, vx, vy, color) {
166     this.w = Math.random() * 4 + 2;
167     this.x = x;
168     this.y = y;
169     this.vx = vx;
170     this.vy = vy;
171     this.alpha = Math.random() * 0.8 + 0.2;
172     this.color = color;
173     this.damping = 0.96;
174     this.gravity = 0.08;
175 }
176
177 Particle.prototype = {
178     move: function () {
179         this.vx *= this.damping;
180         this.vy *= this.damping;
181         this.vy += this.gravity;
182         this.x += this.vx;
183         this.y += this.vy;
184         this.alpha -= 0.01;
185         this.w *= 0.98;
186         return this.alpha > 0 && this.w > 0.5;
187     },
188     draw: function (c) {
189         c.save();
190         c.beginPath();
191         const gradient = c.createRadialGradient(this.x, this.y, 0, this.x, this.y, this.w);
192         gradient.addColorStop(0, this.color);
193         gradient.addColorStop(1, "rgba(0, 0, 0, 0)");
194         c.fillStyle = gradient;
195         c.globalAlpha = this.alpha;
196         c.arc(this.x, this.y, this.w, 0, Math.PI * 2);

```

```

197         c.fill();
198         c.restore();
199     },
200 };
201 </script>
202
203
204 </body>
205 </html>
206

```

```
<!DOCTYPE html>
<html lang="zh">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>烟花动画</title>
  <style>
    body {
      background:
url('https://images-wixmp-ed30a86b8c4ca887773594c2.wixmp.com/f/9c2e9d59
-ee54-490e-bf96-5497f587844f/dgdd80h-c9ed117a-815e-4f0d-9758-b176d26c24
62.png?token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJlcm46YXBwO
jdldMGQxODg5ODIyNjQzNzNhNWYwZDQxNWVhMGQyNmUwIiwiaXNzIjoibWVudC54ZW50
MTg4OTgyMjY0Mzc5YTVMMGQ0MTVlYTBkMjZlMCIsIm9iaiI6W1t7InBhdGgiOiJcL2ZcLzl
jMmU5ZDU5LWVlNTQtNDkwZS1iZjk2LTU0OTdmNTg3ODQ0ZlwwZGdkZDgwC1jOWVkmTE3YS
04MTVlLTRmMGQtOTc1OC1iMTc2ZDI2YzI0NjIucG5nInldXSwiYXVkIjpjbInVybjpzZXJ2a
WNlOmZpbGUuZG93bmxyYWQiXX0.aafHyvc-txx8Fm_dH_FzNriCu4E9Aj7oElBbXJm89h8'
) no-repeat center center fixed;
      background-size: cover;
      overflow: hidden;
      margin: 0;
    }
    canvas {
      background-color: transparent;
    }
  </style>
</head>
<body>
  <canvas id="canvas"></canvas>
  <script>
    window.addEventListener("resize", resizeCanvas, false);
    window.addEventListener("DOMContentLoaded", onLoad, false);
    window.addEventListener("click", launchFireworkOnClick, false);

    var canvas, ctx, w, h, particles = [], fireworks = [];

    function onLoad() {
      canvas = document.getElementById("canvas");
      ctx = canvas.getContext("2d");
      resizeCanvas();
```

```
    window.requestAnimationFrame(updateWorld);
  }

  function resizeCanvas() {
    if (!!canvas) {
      w = canvas.width = window.innerWidth;
      h = canvas.height = window.innerHeight;
    }
  }

  function launchFireworkOnClick(event) {
    const clickX = event.clientX;
    const clickY = event.clientY;
    fireworks.push(new Firework(clickX, h, clickX, clickY));
  }

  function updateWorld() {
    update();
    paint();
    window.requestAnimationFrame(updateWorld);
  }

  function update() {
    for (let i = fireworks.length - 1; i >= 0; i--) {
      if (!fireworks[i].move()) {
        fireworks.splice(i, 1);
      }
    }

    const alive = [];
    for (let i = 0; i < particles.length; i++) {
      if (particles[i].move()) {
        alive.push(particles[i]);
      }
    }
    particles = alive;
  }

  function paint() {
    ctx.globalCompositeOperation = "source-over";
    ctx.clearRect(0, 0, w, h);

    ctx.globalCompositeOperation = "lighter";
```

```
for (let i = 0; i < fireworks.length; i++) {
  fireworks[i].draw(ctx);
}

for (let i = 0; i < particles.length; i++) {
  particles[i].draw(ctx);
}
}

function Firework(sx, sy, tx, ty) {
  this.x = sx;
  this.y = sy;
  this.tx = tx;
  this.ty = ty;
  this.vx = (tx - sx) / 90;
  this.vy = (ty - sy) / 40;
  this.exploded = false;
  this.color =
    "rgb(" +
    (~(Math.random() * 200 + 55)) +
    ", " +
    (~(Math.random() * 200 + 55)) +
    ", " +
    (~(Math.random() * 200 + 55)) +
    ")";
  this.trail = [];
}

Firework.prototype.move = function () {
  if (!this.exploded) {
    this.trail.push({ x: this.x, y: this.y, alpha: 1 });

    if (this.trail.length > 20) {
      this.trail.shift();
    }

    this.vx *= 0.98; // 空气阻力
    this.vy += 0.05; // 模拟重力
    this.x += this.vx;
    this.y += this.vy;
  }
}
```

```
        if (Math.abs(this.x - this.tx) < 5 && Math.abs(this.y -
this.ty) < 5) {
            this.explode();
            this.exploded = true;
        }
        return true;
    }
    return false;
};
```

```
Firework.prototype.draw = function (c) {
    if (!this.exploded) {
        c.save();

        // 发射拖尾
        for (let i = 0; i < this.trail.length; i++) {
            const t = this.trail[i];
            c.beginPath();
            c.arc(t.x, t.y, 2, 0, Math.PI * 2);
            c.fillStyle = this.color;
            c.globalAlpha = t.alpha;
            c.fill();
            t.alpha -= 0.05;
        }

        c.restore();

        // 发射点
        c.beginPath();
        c.arc(this.x, this.y, 4, 0, Math.PI * 2);
        c.fillStyle = this.color;
        c.fill();
    }
};
```

```
Firework.prototype.explode = function () {
    const nParticles = Math.random() * 150 + 200;

    for (let i = 0; i < nParticles; i++) {
        const angle = Math.random() * Math.PI * 2;
        const speed = Math.random() * 5 + 4;
        const vx = Math.cos(angle) * speed;
        const vy = Math.sin(angle) * speed;
```

```

        const particle = new Particle(this.x, this.y, vx, vy,
this.color);
        particles.push(particle);
    }
};

function Particle(x, y, vx, vy, color) {
    this.w = Math.random() * 4 + 2;
    this.x = x;
    this.y = y;
    this.vx = vx;
    this.vy = vy;
    this.alpha = Math.random() * 0.8 + 0.2;
    this.color = color;
    this.damping = 0.96;
    this.gravity = 0.08;
}

Particle.prototype = {
    move: function () {
        this.vx *= this.damping;
        this.vy *= this.damping;
        this.vy += this.gravity;
        this.x += this.vx;
        this.y += this.vy;
        this.alpha -= 0.01;
        this.w *= 0.98;
        return this.alpha > 0 && this.w > 0.5;
    },
    draw: function (c) {
        c.save();
        c.beginPath();
        const gradient = c.createRadialGradient(this.x, this.y, 0,
this.x, this.y, this.w);
        gradient.addColorStop(0, this.color);
        gradient.addColorStop(1, "rgba(0, 0, 0, 0)");
        c.fillStyle = gradient;
        c.globalAlpha = this.alpha;
        c.arc(this.x, this.y, this.w, 0, Math.PI * 2);
        c.fill();
        c.restore();
    },
};

```



```
</script>
```

```
</body>
```

```
</html>
```