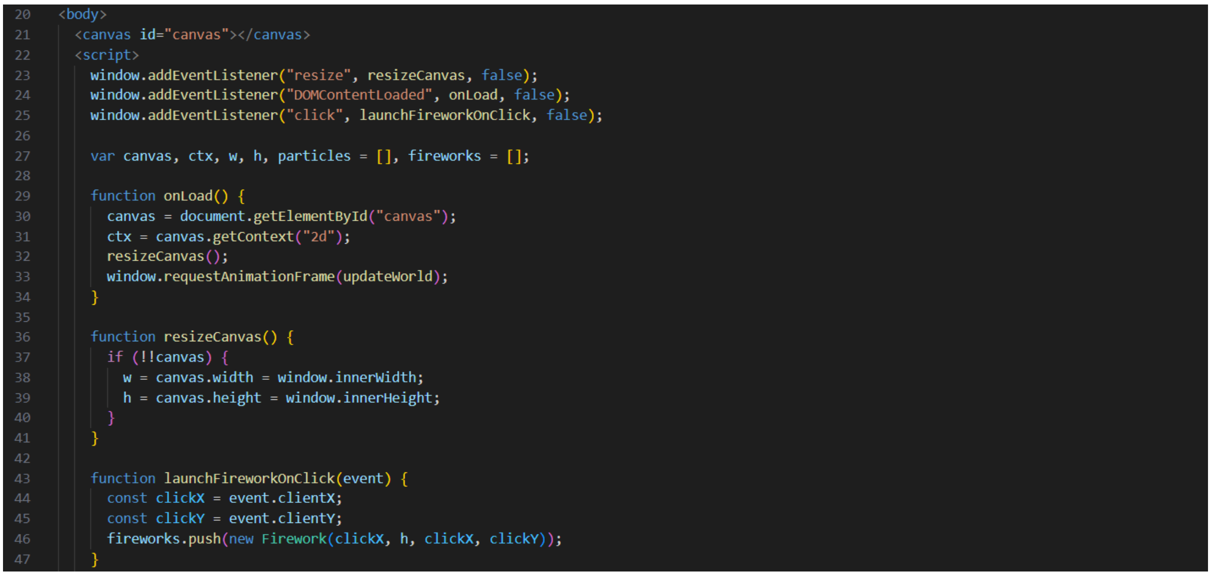
在**PDF** 最下方有源代码可以直接复制并粘贴。

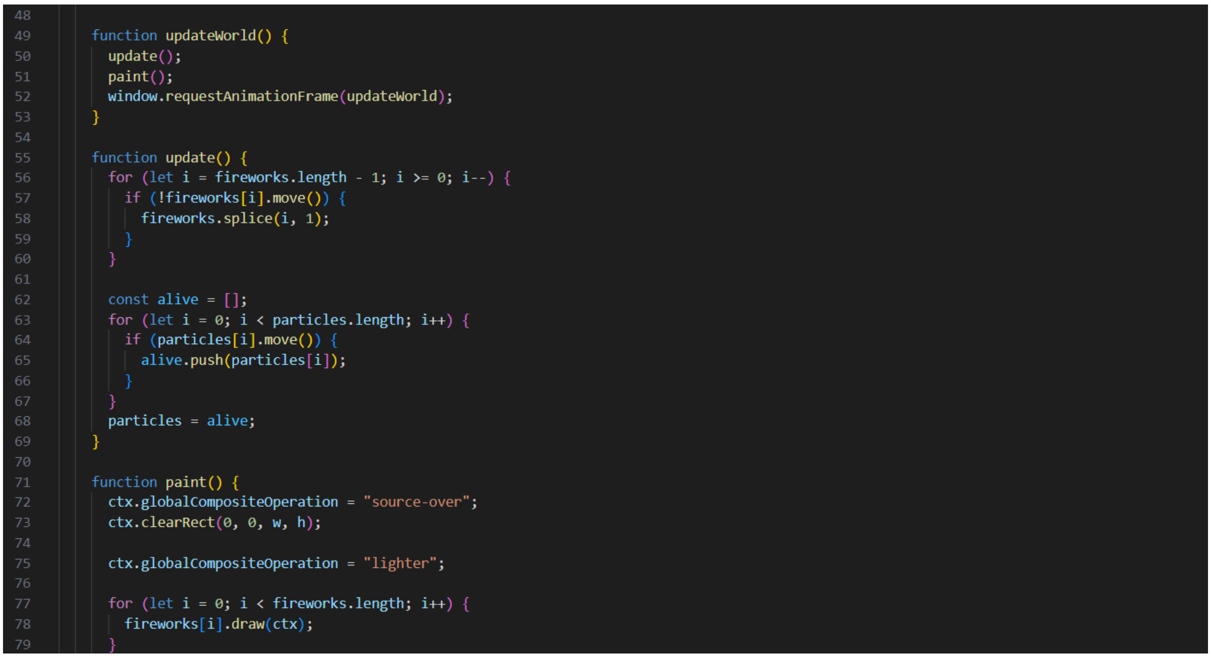
这里是烟花动画链接 https://huoshi111.github.io/yanhuafireworks/yanhua.html

不知道如何将代码写入，可以阅读以下教程

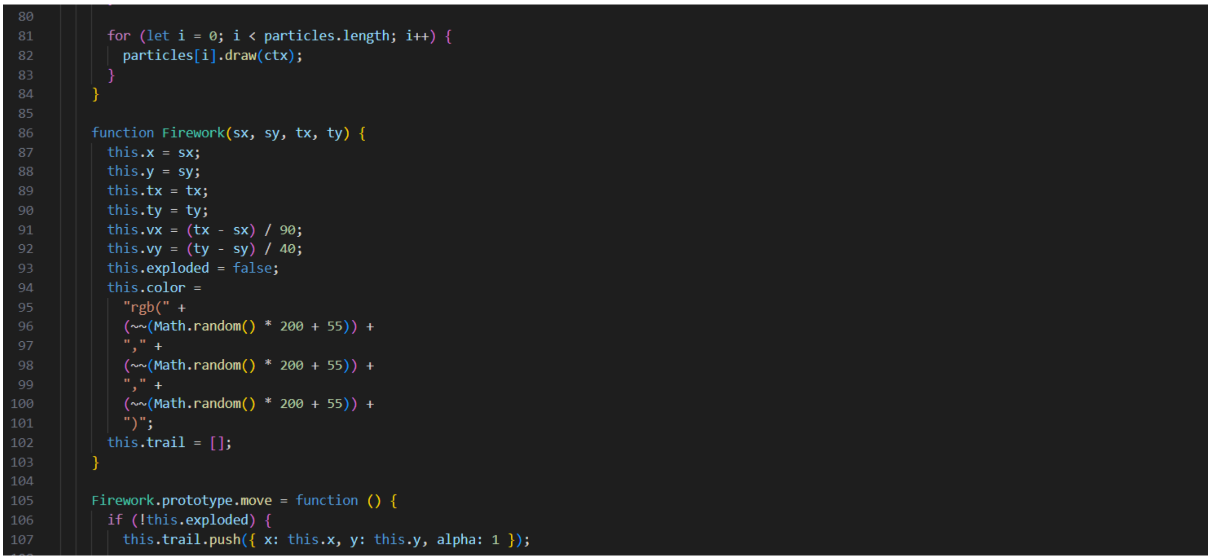
要将代码写入，可以直接在电脑键盘 **WIN + R** 打开执行窗口，在执行窗口里搜索写入**notepad**或中文叫做记事本，在记事本里写入以下代码，在**pdf**最下面有显示源代码，你可以直接复制并粘贴到记事本，粘贴完后，记事本左上角有一个按钮叫做文件下面有一个按钮叫做保存为，点进去，写下一个名字后面记得写一个 **.html** 按保存，这样就有了。



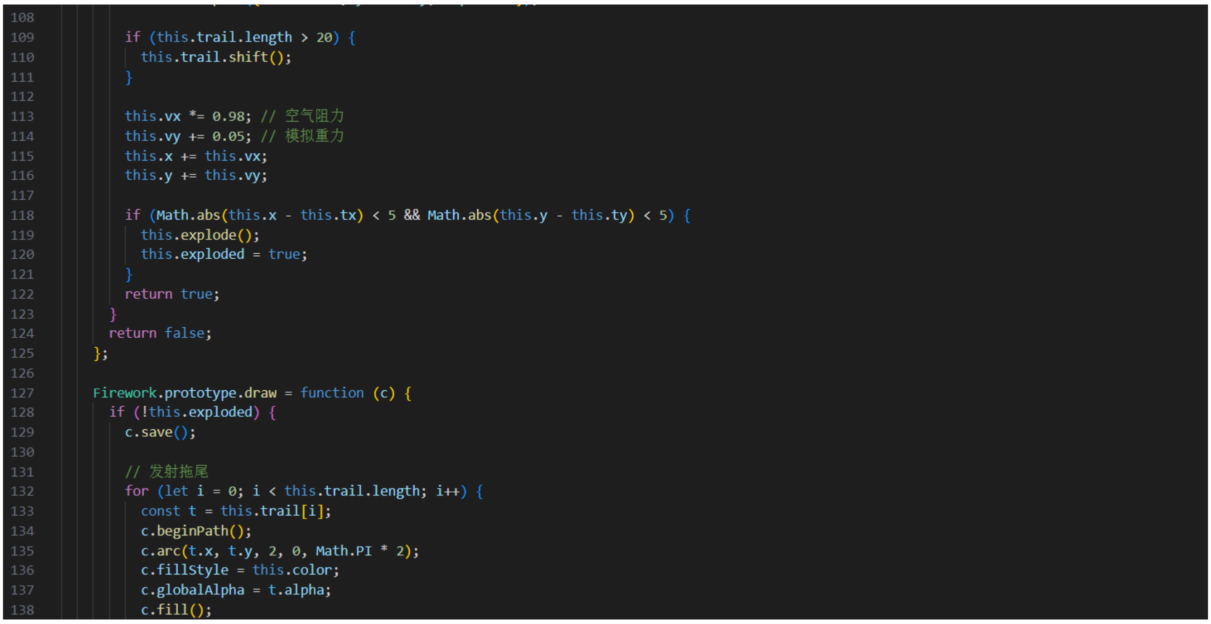
```
<body>
  <canvas id="canvas"></canvas>
  <script>
    window.addEventListener("resize", resizeCanvas, false);
    window.addEventListener("DOMContentLoaded", onLoad, false);
    window.addEventListener("click", launchFireworkOnClick, false);

    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));
    }
```



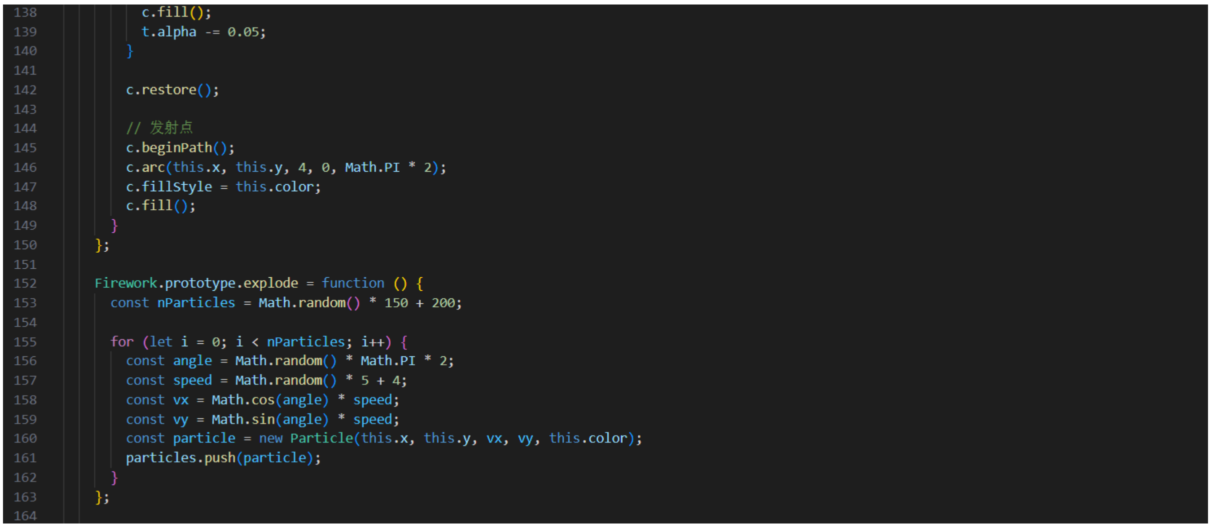
```

    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);
      }
```



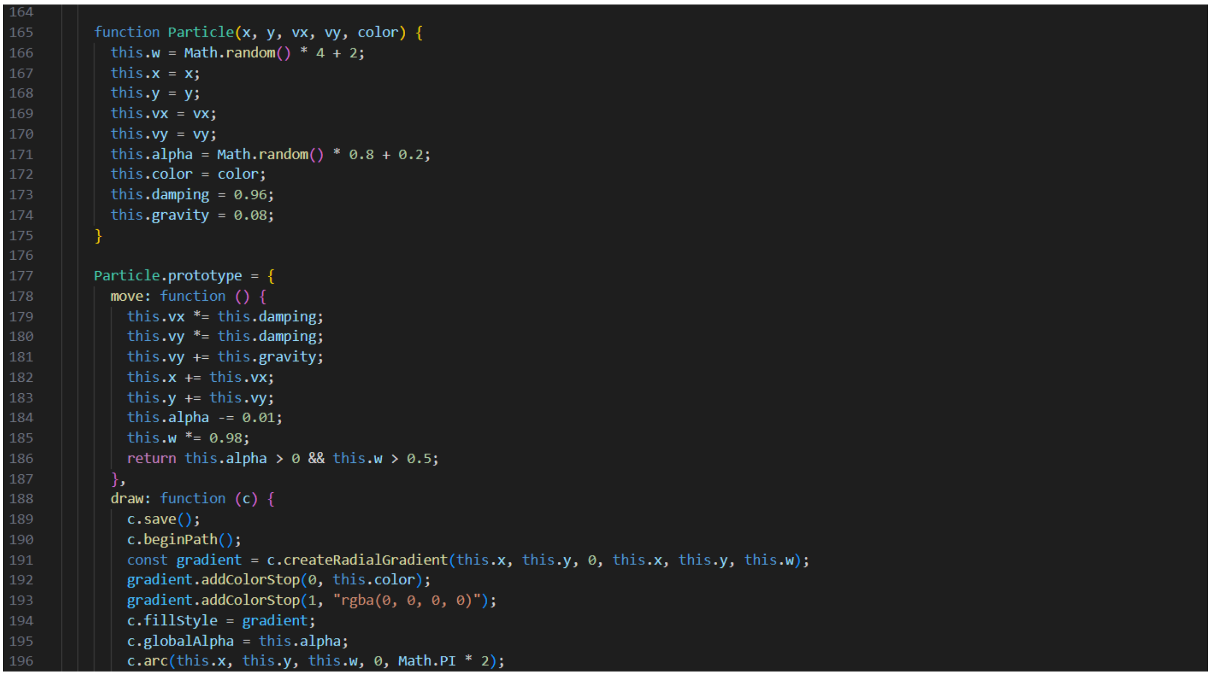
```
      for (let i = 0; i < particles.length; i++) {
        particles[i].draw(ctx);
      }
    }

    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 });
```



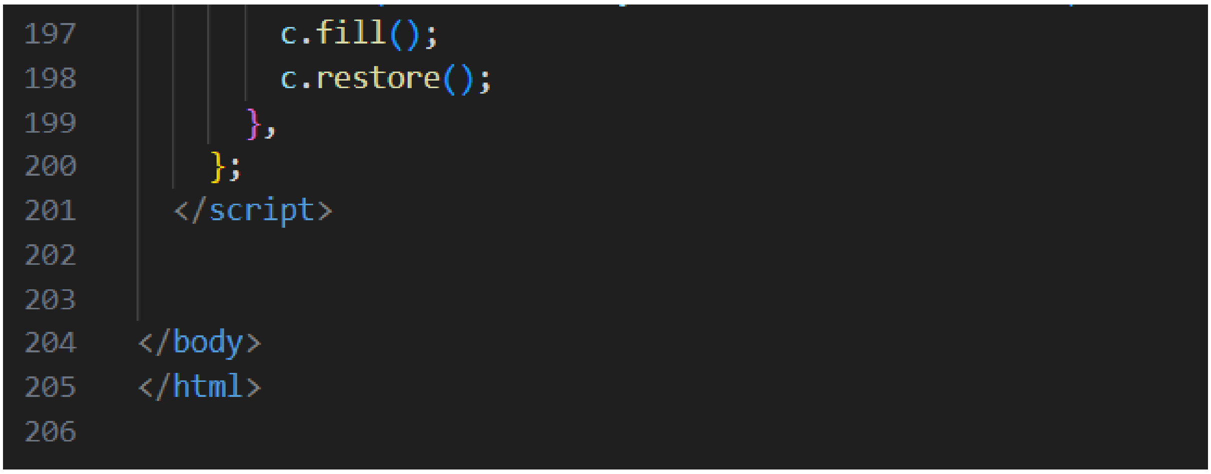
```

        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();
```



```
          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);
      }
    };

```



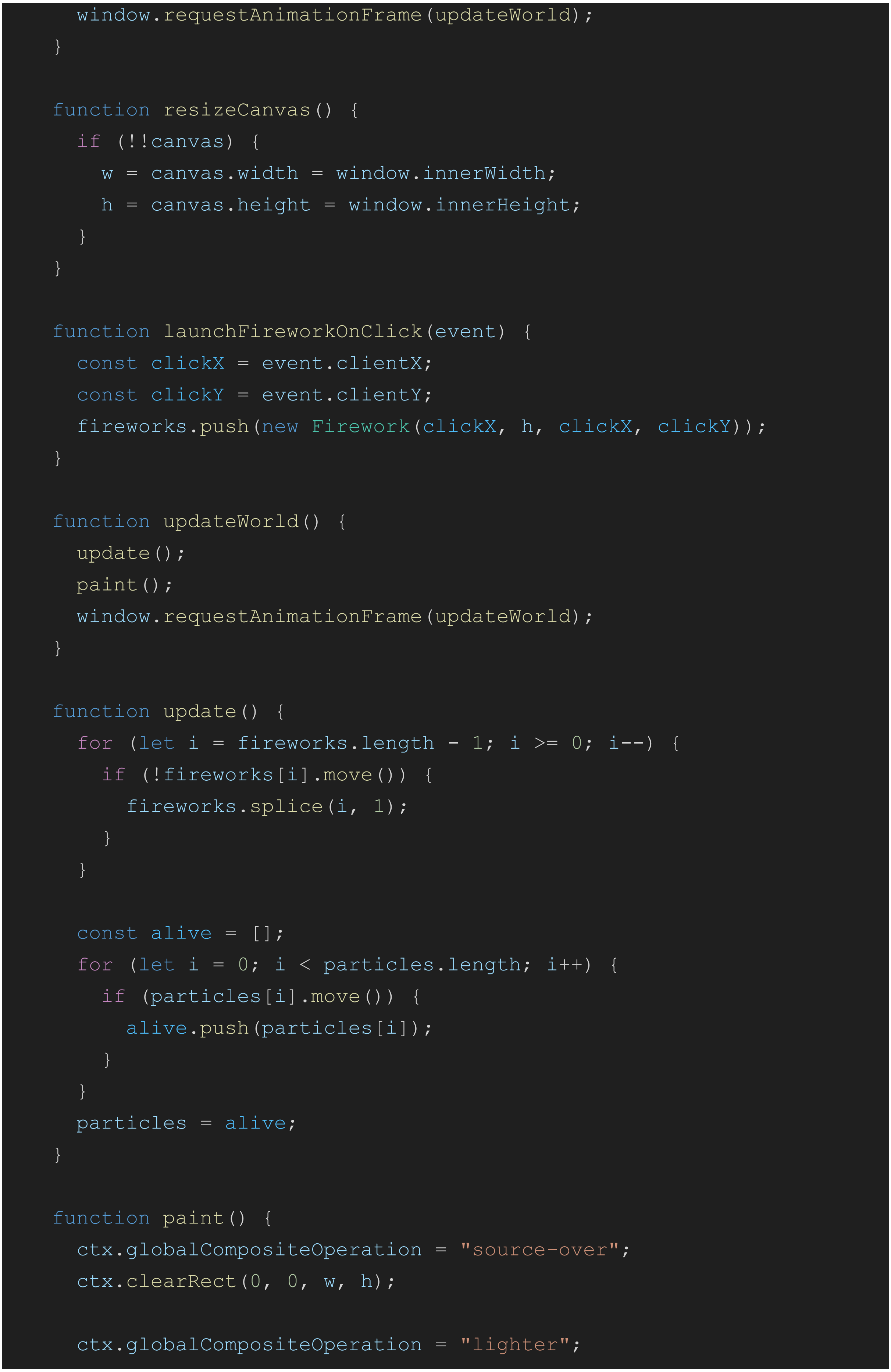
```

    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>

```

在这里长按并复制代码

```
<!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.eyJzdWIiOiJ1cm46YXBwO
jdlMGQxODg5ODIyNjQzNzNhNWYwZDQxNWVhMGQyNmUwIiwiaXNzIjoidXJuOmFwcDo3ZTBk
MTg4OTgyMjY0MzczYTVmMGQ0MTVlYTBkMjZlMCIsIm9iaiI6W1t7InBhdGgiOiJcL2ZcLzl
jMmU5ZDU5LWVlNTQtNDkwZS1iZjk2LTU0OTdmNTg3ODQ0ZlwvZGdkZDgwaC1jOWVkMTE3YS
04MTVlLTRmMGQtOTc1OC1iMTc2ZDI2YzI0NjIucG5nIn1dXSwiYXVkIjpbInVybjpzZXJ2a
WNlOmZpbGUuZG93bmxvYWQiXX0.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>
```