

# canvas图形绘制之星空、噪点与烟雾效果

这篇文章发布于 2016年06月2日, 星期四, 00:34, 归类于 [Canvas](#) 相关。 阅读 47285 次, 今日 7 次 [18 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com>

本文地址: <http://www.zhangxinxu.com/wordpress/?p=5404>

## 一、三合一

三个效果合成一篇文章。

有多个小伙伴问我, 为何不开个公众号, 现在都是移动时代, 你博客文章写好后, 公众号再复制一份, 花不了多长时间, 同时传播方便迅速, 打赏方便快捷, 显然低成本高收益。

从眼前来, 似乎确实如此。

但是, 就我个人而言, 行为和处事准则总是遵循内心的直觉和大方向的指引。说不上具体的道理, 就是觉得, 作品的输出源如果不止一个, 久远来看, 带来的未知损耗一定要大于短期的已知收益。

取巧的事情多慎思而克己, 就好比本文内容, 实际上, 三个不同的canvas效果, 直接分3篇来写, 凑个文章数, 增加点浏览量其实也是无可厚非的。然, 想了想, 有点不像自己的style, 内心真实的自己并不希望自己这么做, 于是, 就3个效果合体为一篇文章。

拒绝小部分的诱惑, 让自己过得更轻松。

本文的3个效果都是源自我最近做的几个真实的项目, 是canvas领域基本入门的一些效果。代码我都专门重新梳理了下, 必要注释也都加上去了, 方便大家的学习。然后, 如果你有不懂的地方, 请不要来问我, 没错, 是不要, 我并不欢迎你找我来交流, 自己一点一点去弄明白。因为, 如果连这么基本的canvas效果都不理解, 我真的也帮不了你什么。倒不是说腾不出时间, 而是腾不出精力, 每天微博私信还有邮箱找我的人还挺多, 实在应接不暇。

## 二、canvas图形效果之旋转星空



图是死的, 效果是活的, IE9+浏览器下, 您可以狠狠地点击这里: [canvas实现的旋转星空效果demo](#)

会看到地球上方会有很多星星在慢慢地绕着地球转啊转, 星星在闪啊闪。

像这类密集型canvas效果, 一般离不开下面这几个关键字: 实例, 随机, 变化与重绘, [requestAnimationFrame](#)。

原理就是：

1. 先画一个位置透明度随机的静态的星星实例对象；
2. 有一个可以改变星星位置和透明度的draw方法；
3. 定时器跑起来，画布不停地清除与绘制，动画效果完成！

原理很简单。

本例子实现的2个难点在于：

#### 1. 月明星稀

星星垂直方向实际上是个伪随机，越靠近地球，星星越密集，而越往上，越稀疏。其算法如下：

```
var getMinRandom = function() {  
    var rand = Math.random();  
    // step的大小决定了星星靠近地球的聚拢程度，  
    // step = Math.ceil(2 / (1 - rand))就聚拢很明显  
    var step = Math.ceil(1 / (1 - rand));  
    var arr = [];  
    for (var i=0; i<step; i++) {  
        arr.push(Math.random());  
    }  
  
    return Math.min.apply(null, arr);  
};
```

很大概率会返回一个数值偏小的值，于是，就可以有“月明星稀”的分布效果了。

#### 2. 圆弧轨迹

其实很简单，我们套用高中时候学的圆方程式就可以了，如下注释截图所述：

```
// 圆的轨迹方程式为:  $(x-a)^2+(y-b)^2=r^2$   
// 因此，已知x，则  $y = \text{Math.sqrt}(r^2 - (x-a)^2) + b$ ;  
// 其中，圆心是(a, b)  
// 在本例子中  
// 圆心坐标是(canvas.width/2, canvas.height - 600 + r);
```

这下题目就简单了，已知a,b, 求y相对于x的函数表达式.....

### 三、canvas图形效果之雪花噪点效果



图是死的，效果这里也是死的，但并不妨碍我们零距离围观，您可以狠狠的点击这里：[canvas实现的噪点效果demo](#)

由于这里是静态的，所以但从这一点来看，似乎比上面星空简单。但是，如果仅仅看绘制一帧，那这里的噪点要比上面的星空要困难些，最大的难点在于对性能的把控。

这么说吧，上面的星空，总共最多就400个点（白色的星星），但是，这里的噪点，例如，demo中画布大小（那我的例子是1920\*500，其中，噪点大小是1像素\*1像素，总共就有960000个绘制点，显然跟400个点完全不是一个数量级的，如果我们真的一个一个绘制下来，肯定，就连Chrome这么牛步的浏览器也会感觉到明显的卡顿，如何优化如何绘制呢？

这就是本例子实现的难点：

1. 数量与性能

我这里是怎么处理的，虽然最终的噪点大小是1920\*500，但是，我们实际上是由N块300\*150的小的像瓷砖一样的小方块拼起来的。换句话说，我实际只绘制了45000个点，比960000显然要小了20倍还不止。

这样，既满足了效果，又保证了性能。

具体实现原理为：

- 1. 创建一个canvas，绘制一个300\*150随机噪点图形；
- 2. 把这里具有噪点的canvas以画布形式在绘制到页面上的大canvas上；

说得canvas绘图，不得不提一下非常常用的一个 `drawImage()` 方法，语法如下：

```
context.drawImage(img,sx,sy,width,height,x,y,width,height);
```

各个参数示意为（网上的描述都是直译，很生涩，我这里重新陈述了下）：

参数
描述
<i>img</i>
用来被绘制的图像、画布或视频。
<i>sx</i>
可选。img被绘制区域的起始左上x坐标。
<i>sy</i>
可选。img被绘制区域的起始左上y坐标。
<i>swidth</i>
可选。img被绘制区域的宽度。
<i>sheight</i>
可选。img被绘制区域的高度。
<i>x</i>
画布上放置img的起始x坐标。
<i>y</i>
画布上放置img的起始y坐标。
<i>width</i>
可选。画布上放置img提供的宽度。（伸展或缩小图像）
<i>height</i>
可选。画布上放置img提供的高度。（伸展或缩小图像）

本例的小的噪点区块就是通过 `drawImage()` 方法被平铺到大的canvas元素上的。

## 四、canvas图形效果之烟雾缭绕效果



图是死的，效果是活的，IE9+浏览器下，您可以狠狠地点击这里：[canvas实现的烟雾缭绕效果demo](#)

本例子，效果看上去要更酷一些，实际上，从技术层面讲，跟上面的星空旋转效果几乎如出一辙，可能还要比星空更简单一些，因为其运动轨迹直来直往，不需要转圈圈。

那为何看上去更酷呢，主要在于感觉烟雾很难去模拟。

没错，烟雾确实很难用代码直接绘制出来，实际上，这里的烟雾，是一个png图片，是使用画笔在PS里绘制导出来的。

旋转星空的例子，我们是使用canvas的 `fillRect` 方法绘制了星星，而本例子，则是使用上面提到的 `drawImage()` 方法把烟雾图片绘制进来了。

其他的位移啊，透明度变化什么的，原理都是类似。

本例子的难点主要在于模拟是否足够真实：

#### 1. 高处不胜寒

越往上，烟雾越淡，实际上就是越靠近上方，透明度越低；

```
// 越靠近边缘，透明度越低
// 纵向透明度变化要比横向的明显
this.alpha = (1 - Math.abs(canvasWidth*0.5 - this.x) / canvasWidth) * (0.7 - Math.abs(canvasHeight*0.5 - this.y) / canvasHeight);
```

#### 2. 缭绕

所谓“缭绕”，就是运动看似不具有规律性。要知道，凡事有轨迹有套路的运动都是有规律性地，你说这烟雾上上下下，左左右右运动太过于规律，效果就会打折扣，但是，真的没有规律又不好通过代码控制运动轨迹。因此，为了搞到一个接近缭绕效果的运动函数，还真是烧了不少脑细胞。

## 五、canvas动效与结语

本文三个例子都是canvas 2D效果，是入门学习非常好的例子。

canvas非常适合实现密集型图形和动画，可以把性能优势给发挥出来，因为就是一块画布渲染；另外一点就是省流量，比方说第2个例子的噪点效果，如果是同样效果1920\*500的png图片，科科，我特意保存了下，286K，1K的代码PK 286K的图片，显然是完爆啊！

canvas还支持3D效果，也就是webGL，亦称3D Canvas graphics，IE11+支持，目前Android 4.\*任意版本都不支持，业内著名的相关库就是threejs了。

不过，我没研究过，也没兴趣，不是我的方向。

好了，就这些，感谢阅读。



《CSS世界》签名版独家发售，包邮，可指定寄语，[点击显示购买码](#)

(本篇完) // 想要打赏？[点击这里](#)。有话要说？[点击这里](#)。



- 小tips: 纯CSS实现蜡烛、火焰以及熄灭后烟雾效果
- canvas实现iPhoneX炫彩壁纸屏保外加pixi.js流体动效
- 基于clip-path的任意元素的碎片拼接动效
- 炫酷H5中序列图片视频化播放的高性能实现
- CSS CSS3实现3D开门动画效果
- canvas 2D炫酷动效的实现套路和需要的技术积累
- 小tips:使用canvas在前端实现图片水印合成
- SVG <foreignObject>简介与截图等应用
- canvas getImageData与任意字符图形点、线动效实现
- 小tips: 在canvas上实现元素图片镜像翻转动画效果
- 使用canvas实现和HTML5 video交互的弹幕效果

分享到:       0

标签: canvas, drawImage, 动效, 噪点, 星空, 烟雾

## 发表评论（目前18条评论）

<input type="text"/>	名称 (必须)
<input type="text"/>	邮件地址(不会被公开) (必须)
<input type="text"/>	网站
<div></div>	
<div>提交评论</div>	

- awzs**说道:  
2017年08月4日 15:37

为什么要在clearRect后面跟上fillRect呢?

[回复](#)
- 夜起叶落**说道:  
2017年07月14日 22:40

赞一个

[回复](#)
- 咕咕鸡**说道:  
2016年08月19日 15:43

没看懂那个星空，那个星星是怎么做的，是所有星星在一个image上，还是每个星星都是image、

[回复](#)

4.

PaulHan说道:

2016年07月15日 11:04

谢谢鑫大大

回复



5.

程满想说道:

2016年07月12日 09:52

你好，我看了关于你的文章，你能否写个关于网页定格的需求demo，谢谢


回复

stanJ说道:

2017年10月20日 16:13

习大大口气

回复




6.

aaron说道:

2016年06月16日 18:43

严谨的说，星星之间的相对位置在短时间内不会变化，除非时间变化的速度很快.....

回复



7.

icewind说道:


2016年06月12日 10:33

$$\text{this.y} = \text{b} - \text{Math.sqrt}(\text{settings.r} * \text{settings.r} - (\text{this.x} - \text{a}) * (\text{this.x} - \text{a})) - \text{this.offsety};$$

按照推导的公式： $y = \text{Math.sqrt}(r^2 - (x-a)^2) + b$ ;

不应该是b加上后面的值再减去offsety么？

回复



8.


anonymous说道:

2016年06月11日 20:44

为什么不直接提供源码呢！

你提供 代码片段存在bugs!

回复



9.

axiba说道:

2016年06月8日 21:31

我可以在博客里用一下你的demo吗，我会标明出处的。。。。

回复



10.

笑一阵癫狂说道:

2016年06月8日 15:21

星空那个例子我看个半天没发现在动 我可能是瞎了

回复

Pandamo说道:

2016年06月12日 09:52

固定盯着一颗星星看，你就看到在转，而且忽隐忽现

回复



11. Pandamo说道:

2016年06月6日 11:34

个人意见，背景再暗点，感觉离地球的星星密了点，随机星星alpha变化速度快点就完美了。

[回复](#)



张 鑫旭说道:

2016年06月6日 22:06

demo这里数目我增加了，为了示意更明显

[回复](#)



12. 路人说道:

2016年06月6日 11:06

```
canvasNoise(canvas, {
  pieceWidth: 300,
  pieceHeight: 150
});
```

300 150是Chrome下创建的canvas默认width height

```
可以 var canvasPiece = function(options) {
var canvas = document.createElement('canvas'), context = canvas.getContext('2d');
canvas.width = options.pieceWidth;
canvas.height = options.pieceHeight;
```

这样尺寸就没有限制了

[回复](#)



13. 萌说道:

2016年06月3日 09:30

大赞~支持

[回复](#)



14. 五说道:

2016年06月2日 16:20

棒

[回复](#)



15. 夏至说道:

2016年06月2日 07:58

学习了。真佩服鑫哥对自己的这种态度。

[回复](#)



最新文章

- » 常见的CSS图形绘制合集
- » 粉丝群第1期CSS小测点评与答疑
- » 分享三个纯CSS实现26个英文字母的案例
- » 小tips: 纯CSS实现打字动画效果
- » CSS/CSS3 box-decoration-break属性简介
- » CSS :placeholder-shown伪类实现Material Design占位符交互效果

- » 从天猫某活动视频不必要的3次请求说起
- » CSS vector-effect与SVG stroke描边缩放
- » CSS ::backdrop伪元素是干嘛用的?
- » 周知: CSS -webkit-伪元素选择器不再导致整行无效

## 今日热门

- » 常见的CSS图形绘制合集 (190)
- » 未来必热: SVG Sprite技术介绍 (119)
- » 粉丝群第1期CSS小测点评与答疑 (115)
- » HTML5终极备忘大全 (图片版+文字版) (93)
- » 让所有浏览器支持HTML5 video视频标签 (86)
- » Selectivizr-让IE6~8支持CSS3伪类和属性选择器 (82)
- » CSS3下的147个颜色名称及对应颜色值 (79)
- » 视区相关单位vw, vh.简介以及可实际应用场景 (76)
- » 写给自己看的display: flex布局教程 (76)
- » 小tips: 纯CSS实现打字动画效果 (76)



## 今年热议

- » 《CSS世界》女主角诚寻靠谱一起奋斗之人 (76)
- » 不借助Echarts等图形框架原生JS快速实现折线图效果 (64)
- » 看, for..in和for..of在那里吵架! (60)
- » 是时候好好安利下LuLu UI框架了! (47)
- » 原来浏览器原生支持JS Base64编码解码 (35)
- » 妙法攻略: 渐变虚框及边框滚动动画的纯CSS实现 (33)
- » 炫酷H5中序列图片视频化播放的高性能实现 (31)
- » CSS scroll-behavior和JS scrollIntoView让页面滚动平滑 (30)
- » windows系统下批量删除OS X系统.DS\_Store文件 (26)
- » 写给自己看的display: flex布局教程 (26)

## 猜你喜欢

- 小tips: 纯CSS实现蜡烛、火焰以及熄灭后烟雾效果
- canvas实现iPhoneX炫彩壁纸屏保外加pixi.js流体动效
- 基于clip-path的任意元素的碎片拼接动效
- 炫酷H5中序列图片视频化播放的高性能实现
- CSS CSS3实现3D开门动画效果
- canvas 2D炫酷动效的实现套路和需要的技术积累
- 小tips: 使用canvas在前端实现图片水印合成
- SVG <foreignObject>简介与截图等应用
- canvas getImageData与任意字符图形点、线动效实现
- 小tips: 在canvas上实现元素图片镜像翻转动画效果
- 使用canvas实现和HTML5 video交互的弹幕效果