

canvas实现一颗树的生长到开花

原创 K-Code 最后发布于2017-08-03 10:20:55 阅读数 3574 ☆ 收藏

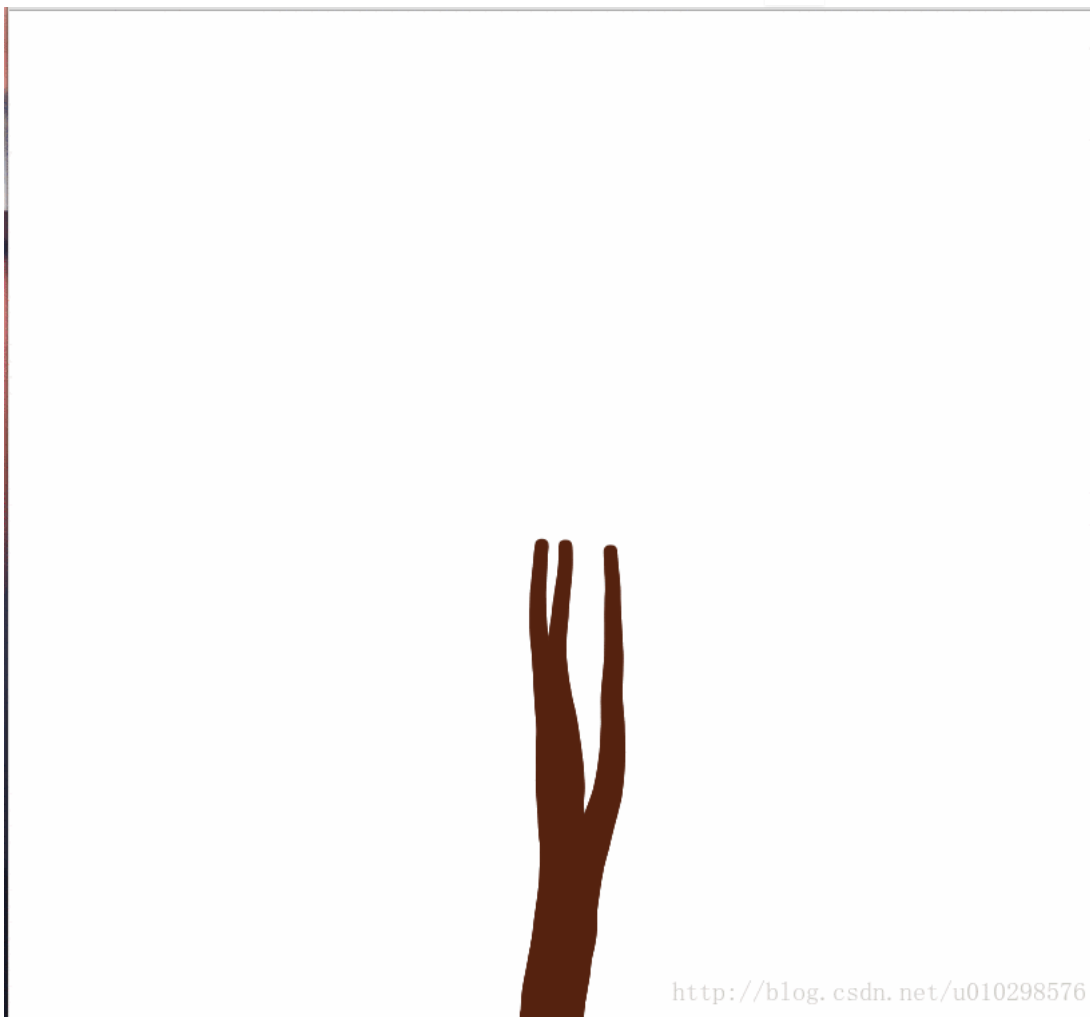
分类专栏: canvas

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/u010298576/article/details/76609244>

最近看了一些很牛逼的特效，有位大神用canvas绘制的 [梦幻大树](#)，深深的震撼了我，[真的很厉害](#)，于是我也准备尝试着去实现，可能脑子不够用，总是：一样的梦幻效果，最终只好退而求其次，实现了下面的这种大树，还算满意吧。

效果图



HTML源码

```
1 <!DOCTYPE html>
2 <html>
3 <head lang="en">
4     <meta charset="UTF-8">
5     <title>生长的树</title>
6     <style>
7         html , body {
8             margin: 0;
9             padding: 0;
```

```
10         width: 100%;
11         height: 100%;
12         overflow: hidden;
13         background-color: #fff;
14     }
15 </style>
16 </head>
17 <body>
18 <canvas id="myCanvas">此浏览器不支持canvas</canvas>
19 <script src="../../js/tree.js"></script>
20 </body>
21 </html>
```



1



tree.js 源码

```
1  /**
2   * Created by 004928 on 2017/8/2.
3   */
4  (function (window) {
5
6      var w = window.innerWidth , h = window.innerHeight ;
7      var ctx = null ;
8      var treeNum = 3 ;
9      var initRadius = 25 ;    // 树干的初始宽度
10     var maxGeneration = 5 ;   // 最多分支的次数
11     var branchArray = null ;  // 树干的集合
12     var flowers = [] ;        // 花的集合
13
14     window.MyRequestAnimationFrame = window.requestAnimationFrame ||
15     window.mozRequestAnimationFrame ||
16     window.webkitRequestAnimationFrame ||
17     window.msRequestAnimationFrame ;
18
19     window.MyCancelRequestAnimationFrame = window.cancelRequestAnimationFrame ||
20     window.mozCancelRequestAnimationFrame ||
21     window.webkitCancelRequestAnimationFrame ||
22     window.msCancelRequestAnimationFrame ;
23
24     /**
25      * 初始化canvas
26      */
27     function initCanvas () {
28         var canvas = document.getElementById("myCanvas");
29         canvas.setAttribute('width' , w);
30         canvas.setAttribute('height' , h);
31         if(canvas.getContext) {
32             ctx = canvas.getContext('2d');
33             initTree();
34             loop();
35         }
36     }
37
38     /**
39      * 初始化树的数量
40      */
41     function initTree () {
42         branchArray = new BranchArray ();
43         for(var i = 0 ; i < treeNum ; i++) {
44             branchArray.add(new Branch(w / 2 , h));
45         }
46     }
47
48     /**
49      * 树干
50      * @param x
51      * @param y
52      * @constructor
53      */
54     function Branch (x , y) {
55         this.x = x ;
```



举报

```
56     this.y = y ;
57     this.radius = initRadius ;
58     this.angle = Math.PI / 2 ; // 树枝的初始角度
59     this.speed = 2.35 ; // 数生长的速度
60     this.generation = 1 ;
61 }
62
63 /**
64  * 生长
65  */
66 Branch.prototype.grow = function () {
67     this.draw();
68     this.update();
69 }
70
71 Branch.prototype.draw = function () {
72     ctx.fillStyle = '#55220F';
73     ctx.beginPath();
74     ctx.arc(this.x , this.y , this.radius , 0 , 2 * Math.PI);
75     ctx.fill();
76 }
77
78 /**
79  * 更改数的高度以及扭曲度
80  */
81 Branch.prototype.update = function () {
82
83     // 计算树干每次的扭曲角度，因为树一般不是笔直生长的，都会有不规则的扭曲
84     this.angle += random( -0.1 * this.generation / 2 , 0.1 * this.generation / 2 );
85
86     var vx = this.speed * Math.cos(this.angle);
87     // 因为初始角度设置为Math.PI ， 所以vy要取负数
88     var vy = - this.speed * Math.sin(this.angle);
89
90     if(this.radius < 0.99 || this.generation > maxGeneration) {
91         branchArray.remove(this);
92     }
93
94     this.x += vx ;
95     this.y += vy ;
96
97     this.radius *= 0.99 ;
98
99     if(this.radius >= 0.9) {
100         // 计算当前是第几代分支
101         var g = (maxGeneration - 1) * initRadius / (initRadius - 1) / this.radius + (initRadius - maxGeneration) / (initRadius - 1);
102         if( g > this.generation + 1 ) {
103             this.generation = Math.floor(g) ;
104             // 随机创建分支
105             for(var i = 0 ; i < random(1,3) ; i++) {
106                 this.clone(this);
107             }
108         }
109     }
110 }
111
112
113 /**
114  * 创建分支
115  * @param b
116  */
117 Branch.prototype.clone = function (b) {
118     var obj = new Branch(b.x , b.y);
119     obj.angle = b.angle ;
120     obj.radius = b.radius ;
121     obj.speed = b.speed;
122     obj.generation = b.generation;
123     branchArray.add(obj);
124     // 如果当前分支次数大于3则创建花，这样可以让花在树的顶端显示
125     if( b.generation > 3 ) {
126         flowers.push(new Flower(b.x , b.y));
```



1



举报

```
127     }
128 }
129
130 function BranchArray () {
131     this.branchs = [];
132 }
133
134 /**
135  * 添加树干到集合中
136  * @param b
137  */
138 BranchArray.prototype.add = function (b) {
139     this.branchs.push(b);
140 }
141
142 /**
143  * 从集合中移除树干
144  * @param b
145  */
146 BranchArray.prototype.remove = function (b) {
147     if( this.branchs.length > 0) {
148         var index = this.branchs.findIndex(function (item) {
149             return b === item ;
150         })
151         if(index != -1) {
152             this.branchs.splice(index , 1);
153         }
154     }
155 }
156
157 /**
158  * 花
159  * @param x
160  * @param y
161  * @constructor
162  */
163 function Flower (x , y) {
164     this.x = x ;
165     this.y = y ;
166     this.r = 1 ; // 花瓣的半径
167     this.petals = 5 ; // 花瓣数量
168     this.speed = 1.0235 ;// 花的绽放速度
169     this.maxR = random(3 , 7); // 花的大小
170 }
171
172 /**
173  * 花朵开放（通过改变花的半径实现开放的效果）
174  * @param index
175  */
176 Flower.prototype.update = function (index) {
177     if(this.r == this.maxR) {
178         flowers.splice(index , 1);
179         return ;
180     }
181     this.r *= this.speed ;
182     if(this.r > this.maxR) this.r = this.maxR ;
183 }
184
185 /**
186  * 绘制花朵
187  */
188 Flower.prototype.draw = function () {
189     ctx.fillStyle = "#F3097B" ;
190     for(var i = 1 ; i <= this.petals ; i++) {
191         var x0 = this.x + this.r * Math.cos( Math.PI / 180 * (360 / this.petals) * i) ;
192         var y0 = this.y + this.r * Math.sin( Math.PI / 180 * (360 / this.petals) * i) ;
193         ctx.beginPath();
194         ctx.arc(x0 , y0 , this.r , 0 , 2 * Math.PI) ;
195         ctx.fill();
196     }
197     ctx.fillStyle = "#F56BC1";
198     ctx.beginPath();
```



1



举报

```
198     ctx.arc(this.x , this.y , this.r / 2 , 0 , 2 * Math.PI) ;
199     ctx.fill();
200 }
201
202 function random (min , max) {
203     return Math.random() * (max - min) + min ;
204 }
205
206 /**
207  * 循环遍历所有树干和花，并调用更新和draw方法，实现动画效果
208  */
209 function loop () {
210     for(var i = 0 ; i < branchArray.branchs.length ; i ++ ) {
211         var b = branchArray.branchs[i];
212         b.grow();
213     }
214     var len = flowers.length ;
215     while (len --) {
216         flowers[len].draw();
217         flowers[len].update();
218     }
219     MyRequestAnimationFrame(loop);
220 }
221
222 window.onload = initCanvas;
223
224 })(window)
```



1



原理

- 1.生长: 因为树是越往上树干越细，我们通过绘制圆，通过无数的圆叠加来当做树的躯体，逐渐缩小圆的半径，即可实现树越往上越小的效果
- 2.扭曲: 其次树的生长不是笔直的，是扭曲的，所以通过三角函数随机产生角度值，然后计算x,y轴的偏移量，这样就能实现树的扭曲
- 3.分支: 通过双曲线方程 $y = -a/x + b$ 去控制什么树在什么时候去产生分支（待会详解）；
- 4.开花: 因为花一般都是长在末梢，而我则是显示在最后几个分支的结点以及末梢进行展示（这个开花地点随意，不必纠结）

生长

树的生长变细，主要通过改变树干的半径即可， $this.radius = 0.99 * \text{每次递减}$;

扭曲

关于树的扭曲，这个是重点，因为树枝延伸和扭曲都是不规则的，在 *Branch.prototype.update* 中

```
1 | this.angle += random( -0.1 * this.generation / 2 , 0.1 * this.generation / 2 );
```

在区间[-0.1,0.1]中随机产生角度，通过三角函数计算x,y轴偏移量

```
1 | var vx = this.speed * Math.cos(this.angle);
2 | // 因为初始角度设置为Math.PI , 所以vy要取负数
3 | var vy = - this.speed * Math.sin(this.angle);
```

vx的正负，决定了树的扭曲，vy的正负，决定树是向上生长还是向下

你可能觉得奇怪树为什么会向下生长，首先要理解我们这里将树抽象成了一个一个的圆

无数个圆连接在一起才组成树干，这个要能理解，然后，有些树的枝干有时是会往下生长一段，然后又往上生长的

这个生长的方向我们是不能控制的，并不是所有树都一直向上生长的，所以当vy为负值时就可以实现树枝

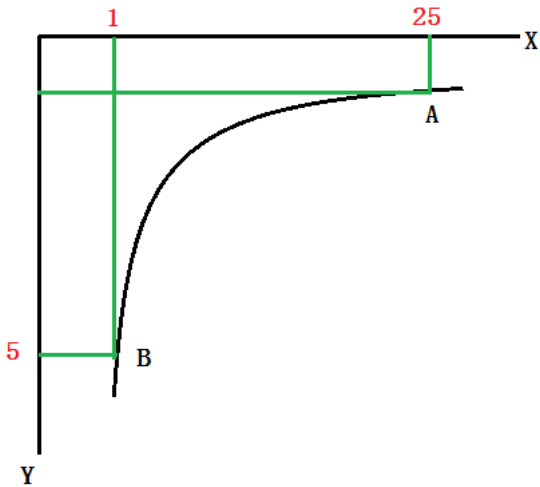
向下生长的情况，当然这个也是随机的，也正是我们想要的效果。



举报

分支

对于在什么时候应该让树去产生分支，我之前的想法是规定一段树干的长度，然后每次计算当前位置距离上一个分支点的距离是否大于我规定的长度，然后产生分支，但是这样就会看到每节分支之间的长度是一样的看起来不美观，比较死板，最终使用双曲线方程 $y=-1/x$ 去控制因为双曲线的走势是先快后慢^^而树的生长也是越往后分支越多,可能你会奇怪曲线是先快后慢，树分叉是先慢后快的，不符合自然规律，别着急，下看下图:



<http://blog.csdn.net/u010298576>

从图中可以看见，X轴表示树干的粗细，Y轴表示分支的次数，当树干越来越细的时候，X轴变小是不是Y轴就越来越大，且是先慢后快，这样就符合我们的需求了。因此上面计算分支的次数公式就由此而来：

```
1 | var g = (maxGeneration - 1) * initRadius / (initRadius - 1) / this.radius + (initRadius - maxGeneration) / (initRadius - 1) ;
```

得到分支次数，比较是否大于当前分支，于是便随即产生分支数量，详见 `Branch.prototype.clone` 也就是创建一个树干添加到BranchArray中

开花

至于开花的逻辑就比较简单，主要记录花朵开放的位置，然后绘制出来便可，至于花朵的形状，详见 `Flower.prototype.draw` 方法这里我就不做过多介绍了，比较简单，而花朵的开放效果，就是改变花朵的半径

一颗大树从生长到开花所有的逻辑步骤大概这四步，其中最主要的逻辑是扭曲，为了这个扭曲我想了好多天，苦闷死我了。最后通过循环调用 `loop` 方法，遍历所有树干以及花朵，即可实现动态效果。


Github 源码下载

点赞 1

收藏

分享

...



K-Code

发布了15 篇原创文章 · 获赞 16 · 访问量 5万+

私信



程序猿不会英语怎么行？英语文档都看不懂！

不死记，不硬背，用公式巧学英文→



举报



想对作者说点什么

<div>canvas demo-通过递归生成树图</div> <div>效果 html tree</div> <div><div>一棵开花的树(源码)</div><div>MFC画的一颗开花的树!艺术与理性的结合!</div></div> <div>canvas实现吃豆人小demo_canvas,html5_weixin_44049725..._CSDN博客</div> <div>...canvas画布元素 制作 动态花朵动画_菜菜菜菜鸟の成长记-CSDN博客</div> <div><div>HTML5 Canvas花开背景动画特效.zip</div><div>HTML5 Canvas花开背景动画特效是一款适合用作网页动态背景素材下载。</div></div> <div><div></div><div>程序员学习编程的36个网站，超实用！ 编程基础课程</div></div> <div>js通过canvas实现动画效果 - m0_37907835的博客 - CSDN博客</div> <div><div>【效果】html5给小树浇水，然后树长大的效果</div><div>用到的 浇水 /* animation */.a-swing(-webkit-animation:1s ease;-moz-anim...</div></div> <div><div>JavaScript 控制 （改变） canvas(画布)的大小</div><div>前段时间在论坛看到有人问如何自由改变画布大小,闲来无事,就花了点时间解决,在CSDN博客中没有相关博文,于是顺...</div></div> <div><div>用js实现开花效果</div><div>JavaScript实现很漂亮的开花特效 - www.unimagical.com>var numberofblossoms=15 //数量var openblossom...</div></div> <div><div>htm5 用canvas递归画树</div><div>今天看了一篇用canvas画树的文章，代码简单，于是便按照作者的讲述去实现了这个例子，演示网址如下点击打开链...</div></div> <div><div>canvas 实现玫瑰花</div><div>html<canvas id="c"></canvas>jsvar b = document.body; var c = document.getElementsByTagName...</div></div> <div><div></div><div>订单管理系统 智能订单管理系统</div></div> <div><div>[Canvas]游戏增加怪物爆炸效果，改善箭头形状</div><div>请点击此下载代码并用浏览器打开试玩。图例：代码：<!DOCTYPE html><html lang="utf-8"><meta http-equiv=...</div></div> <div><div>利用canvas绘画二级树形结构图</div><div>上周需要做一个把页面左侧列表内容拖拽到右侧区域，并且绘制成关系树的功能。看了设计图，第一反应是用canvas...</div></div> <div><div>html5中怎么在canvas中画一个二叉树？</div><div>是否用到递归？ ？ ？ ？ ？ ？ 具体怎么实现？ ？ ？ ？ ？ ？ 谢谢</div></div> <div><div>Canvas梦幻大树生长动画特效.zip</div><div>Canvas梦幻大树生长动画特效是一款美丽酷炫的大树生长动画效果。</div></div> <div><div>如何使用d3js画分帧画一棵树</div><div>最近用d3js写了一个把树可视化的工具，但是遇到一个问题，当树的结点非常多的时候，加载整棵树会卡一段时间， ...</div></div> <div><div>HTML5 canvas画的摇摆的树</div><div>Tree cssass.com提醒您： ie9以下用户请一边惭愧去吧 var con=document.getElem...</div></div> <div><div>canvas动态画出视图树</div><div>动态添加/删除视图树节点</div></div>
--

👍

1

🔗

💬

📖

☆

📱

<

>

👍

🔊

🚩

用canvas画一颗树顶有雪的松树 今天下午没事，于是练习画了一颗松树，我的方法比较简单粗暴，不知道有没有更优雅画法，等我找到了再贴上来。... 博文 来自: dream0726的博客	1		581
求如何实现这种树状图 如题 请问这种树状图是如何做的!!!!		论坛	
求一开源可绘制树形结构web控件 用户点击节点，读取数据库绘制节点下一级节点，根据数据库交互绘制一个树形拓扑结构，不知道有没有这样的web控件		论坛	
HTML5 canvas 绘图例子(圣诞树) 《HTML5高级程序设计》教程中 关于讲解Canvas 的案例源代码。包括曲线，直线，阴影，渐变等		02-03	下载
如何用canvas画出这种图片的类似效果? -		论坛	
HTML中的<canvas>分形树 如图，如果知道一个线段的长度L，和它与X轴的夹角，那么我们可以求得线段的另一个顶点的坐标 (L*cos (t) ... 博文 来自: humantic的专栏	赏	1708	
H5网页Canvas画布绘图 图片文字合成 一个很简单的小例子，直接贴代码，代码中都有注释，下面有demo链接。<html><head> <meta name="viewpo... 博文 来自: 季小沫的博客		1万+	
纯js脚本1k大小的3D玫瑰 西班牙牛逼程序员写的： with(m=Math)C=cos,S=sin,P=pow,R=random;c.width=c.height=f=500;h=-250;func... 博文 来自: germmy-神一样的...		219	
自动部署 web ui 操作 自动部署 web ui 操作 #!/usr/bin/env python#encoding=utf-8from bottle impor...		11	博文
在中国程序员是青春饭吗? 今年，我也32了，为了不给大家误导，咨询了猎头、圈内好友，以及年过35岁的几位老程序员.....舍了老脸去揭人家... 博文 来自: 启舰		15万+	
深析Synchronized关键字（小白慎入，深入jvm源码，两万字长文） 从jvm层面解析synchronized，看完绝对可以超越绝大多数人		7095	博文 来自: Java新生代
大学四年，我决定把Java学习过的书籍都分享一遍 给岁月以文明，而不是给文明以岁月，技术人读书我觉得很有必要，那这份书单的大部分书我觉得对您有用。... 博文 来自: 敖丙		1万+	
程序员请照顾好自己，周末病魔差点一套带走我。 程序员在一个周末的时间，得了重病，差点当场去世，还好及时挽救回来了。...		8万+	博文 来自: 敖丙
卸载 x 雷某度! GitHub 标星 1.5w+，从此我只用这款全能高速下载工具！ 作者 Rocky0429来源 Python空间大家好，我是 Rocky0429，一个喜欢在网上收集各种资源的蒟蒻...网上资源眼... 博文 来自: Rocky0429		16万+	
Java C语言 Python C++ C# Visual Basic .NET JavaScript PHP SQL Go语言 R语言 Assembly language Swift Ruby MATLAB PL/SQL Perl Visual Basic Objective-C Delphi/Object Pascal Unity3D			

没有更多推荐了, 返回首页



K-Code

TA的个人主页 >

原创15

粉丝20

获赞16

评论24

访问5万+

等级: 博客 3

周排名: 14万+

积分: 690

总排名: 11万+

勋章: 



举报

已关注

私信



宿舍管理系统

最新文章

基于jQuery的前端国际化方案

vue+apicloud搭建的一个移动端开发框架

通过回车键提交form表单时需要注意的问题

JS原型链继承

详解JS中函数、函数原型和函数实例之间的关系

分类专栏



Android

1篇



JavaScript

6篇



web前端

6篇



canvas

2篇



移动端

1篇

归档

2019年1月

7篇

2018年5月

1篇

2017年8月

3篇

2017年6月

1篇

2016年12月

1篇

2016年11月

1篇

2016年1月

1篇

展开

热门文章

Web后台管理系统框架模板

阅读数 12648

动态渲染 Element UI el-menu 组件

阅读数 11631

Jquery实现弹幕效果

阅读数 6850

Vue 监听列表item渲染事件

阅读数 3948

canvas实现一颗树的生长到开花

阅读数 3561

最新评论



1



















举报

基于jQuery的前端国际化方案

u010298576: [reply]qq_36258899[/reply]哥们你自己试一下不就知道了

基于jQuery的前端国际化方案

qq_36258899: 大哥，请问如果强制刷新他的话，会不会闪出初始化文字，或者空白

vue+apicloud搭建的一个...

biekendiesb: [reply]u010298576[/reply]好吧，我看了你的另一篇文章，以为说的就是这中逻辑 ...

vue+apicloud搭建的一个...

u010298576: [reply]biekendiesb[/reply]我这个不是vue项目，只是用了vue.js，跟你说的不是 ...

vue+apicloud搭建的一个...

biekendiesb: [reply]u010298576[/reply]需要在ApiCloud studio下安装vue插件吗，这个工具 ...

 QQ客服

 kefu@csdn.net

 客服论坛

 400-660-0108

工作时间 8:30-22:00

[关于我们](#) [招聘](#) [广告服务](#) [网站地图](#)

京ICP备19004658号 经营性网站备案信息

 公安备案号 11010502030143


©1999-2020 北京创新乐知网络技术有限公司


网络110报警服务


北京互联网违法和不良信息举报中心


中国互联网举报中心 家长监护


版权与免责声明 版权申诉


 1






















举报