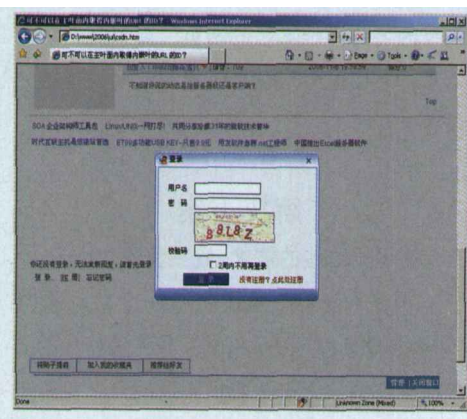


```
<a href="login.aspx" onclick="MzModalDialog
.open(this.href,
{ scroll: 'no', dialogWidth: '360px',
dialogHeight: '200px',
title: '登录', onunload: function(e)
{document.getElementById('divId')
.innerHTML=MzModalDialog
.returnValue}});
return false;">登录</a>
```

这个类的第一个参数是<iframe>要加载的页面地址，第二个参数是一个 JSON 对象，作为对话框的配置来说 JSON 是再好不过了。



经验分享

对话框模式现在之所以没有流行起来，原因其一是实现起来比较繁琐，其二是大部分人还没有体会到它会给网页的用户体验带来什么样的良好改进。Web 2.0 的核心就是与用户的互动，而互动的界面需要方便快捷的操作，诸如选中一个多选框或者切换一下<select>选项就要刷新一次页面的做法终归会被用户所淘汰，所以 AJAX 开始逐年火爆，而随着访问者对网站的用户体验的进一步提升，对话框模式将会在不久的将来大面积流行起来。■

作者简介



黄方荣，网名 meizz 梅花雪，业余专长是动态网页开发，有 6 年 Web 开发经验。长期活跃于 CSDN 论坛，现执掌

CSDN 的 UI 设计部。



对 JavaScript 技术的熟悉程度如幻术者并不多见。选择这篇文章是为了告诉读者，采用 JavaScript 同样能够制作出精美的展现层效果。当然，在此基础上除了要考虑在表现层进行逻辑操作时，需要注意的算法的使用以外，还需要对脚本语言的性能有充分了解。完成界面设计并不是一定要采用某种技术，而是在不同的技术体系下的权衡。

对很多从事企业级应用开发的程序员来说，JavaScript 是一种颇为神秘的语言。这种语言由浏览器解析，可以实现很复杂的展现功能，但在实际中使用甚少。在通常环境下，一个网站体系中的 JavaScript 并非必要，最早的浏览器并不支持 JavaScript，只需使用纯静态的 HTML，也能建立功能完备的网站。对于页面开发人员来说，使用 JavaScript 的时候估计也就是用于数据合法性检查而已。近年来，随着 AJAX 概念崛起，JavaScript 的使用有所增加，不过除了 Web 邮箱等少数工具型的网站之外，其作用仍然是辅助性的。

实际上，JavaScript 作为最流行的脚本语言，功能是非常强大的，笔者就曾经利用 JavaScript 来模拟星际争霸。大家都知道，星际争霸是即时战略游戏史上的里

程碑，在玩家心目中的地位是不言而喻的，泡在各种各样的网吧里玩星际曾经是笔者生活的一部分。仅凭一腔热血，我就着手用 JavaScript 来开发星际，这是一次艰难的历程，充满了挫折，也享受到了乐趣，最后能够完成，却是开始时我没有奢望到的。

JavaScript 星际源码最早发布于无忧脚本的论坛里，得到了很多网友的好评，让我感动之余，也唯有更加努力。通过这次开发，我的 JavaScript 开发技术有了很大提高，还认识了许多喜欢 JavaScript 开发的朋友。JavaScript 星际现在的存放地址是 <http://www.script8.com/works/sc/index.htm>，有兴趣的朋友欢迎参与讨论。

寻径算法是即时战略游戏的基础，也是运算量最大的部分。我没有使用常见的 A* 算法，因为对脚本来说，这种算法的开销太大，而是自主开发一种新的算法，称之为交点法。交点法的特点是线性寻径，运算量比较小，缺点是不能保证结果为最短路径。交点法的基本思路简述如下：

- 从起点到终点画直线，与障碍物相交

用 JavaScript 重建 星际争霸

□ 文 / 幻宇

时,出现对应的穿入点和穿出点:

- 从穿入点的两个方向同时绕行障碍物,选择先到达穿出点的路线作为前进路线,这样就得到了一条比较原始的路线;

- 对路线进行优化,判断两点是否形成通路,删除冗余点,得到最终的路径。

寻径时大量使用了判断是否障碍点的运算,一般的做法是遍历数组,分别比较才能得到结果。我把障碍点序列变成一个长字符串,通过字符串包含关系就可以判断出当前点是否障碍点,这样减少了很多运算。

JavaScript 星际所实现的,相当于星际争霸的开场部分,即采矿、制造建筑物和生产士兵,战斗部分略有涉及。在编码之前,就需要有许多准备工作,光是图片的制作就非常繁琐,需要万分耐心。另外,我还用 HTA 技术开发了一个地图编辑器,用于生成地图数据。

游戏里的控制面板比较精简,缩略地图和选择信息集成到右上角的一个小面板里,其他部分都属于游戏场景。每个移动单位有 8 个方向,以枪兵为例,需要用 24 张图片来表现站立和走动各种姿态。这些图片都集成在一

张大图里,根据走动或站立情况,显示其中的对应部分,比如在走动时,判断出走动方向,一边移动枪兵位置,一边三张图片轮显,表现出走动的情形。

游戏初始化时,载入地图数据,根据不同的建筑物属性,一方面生成场景,另一方面生成障碍物列表,用于寻路之用。默认情况下,地图上有几个矿工,圈选后,点击目标可以自主行走,根据游戏要求,如果点击到的是矿石,就会在矿区和总部间来回走动,每次往返都会增加矿藏量。

至于哪一种建筑物能生产什么兵,每个兵需要多少资源,建筑物的生产顺序是什么,这些相对来说都比较容易,无非是做更多的图片处理和更多的逻辑判断而已,限于篇幅就不详细介绍了。

在游戏的设计过程中,开发人员不用担心功能的实现,因为 JavaScript 已经非常完善了,可以随心所欲地进行各种运算。使用绝对定位的机制,可以轻松创建界面,再利用时钟模拟多线程,实时移动图片,就可以表现动画了。尽管 JavaScript 的功能非常强大,但是它也同样存在问题,目前看来,其缺点主要体现在速度和性能上。因为 JavaScript 作为一种脚本语言,其计算性能先天不足,同时浏览器也不支持 DirectX 和 OpenGL 等硬加速,动画能力无法另人满意。只有通过减轻运算量,合理分配动画资源,才能保证游戏的平滑运行。JavaScript 星际的开发,正是处处遵循了这个原则,能简则简,能省则省,非常节约地使用各种资源,才最终得以实现。

值得注意的一点是,千万不要在游戏中使用滤镜技术,特别是动态滤镜。滤镜渲染时会占用大量 CPU 资源,如果在 CPU 占用达到 80% 以上时,页面上的游戏就会感觉有明显的迟滞。这一点和 Flash 做的同类程序有明显区别,即使 Flash 程序占用 CPU 接近 100%,仍然能保持相当的平滑。

然而,对于浏览器的游戏开发我仍然建议以 Flash 为主,毕竟对于展现技术来说,Flash 的表现能力要比 JavaScript 好。与之相比,JavaScript 的好处是可与网站无缝连接,不需要安装任何插件。如果你对于页面上的游戏开发很有兴趣,采用 JavaScript 开发也未尝不可,通过一次这样的宝贵经历必将让你成为页面展现技术的高手。■

作者简介



幻宇,原名杨兴祥,重庆大学毕业。有多年的 Web 客户端开发经验,熟悉 ActionScript 和 JavaScript。对 Web 游戏开发很有兴趣。代表作有 JavaScript 星际,JavaScript 象棋和 XFlash 引擎等。个人爱好是收藏小人书、徒步旅行和看电影。