

HTML 5

完美游戏开发

【白俄罗斯】Egor Kuryanovich 等 著

阳飞 海通 李碎伍 译



无论您是寻找编程秘密来进一步完善自己的游戏的专家，或是寻找灵感和坚实的编程技巧的游戏初学者，本书都给你最期待的答案：

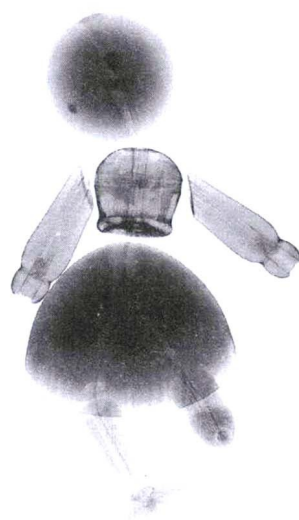
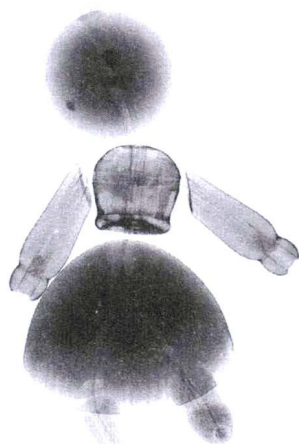
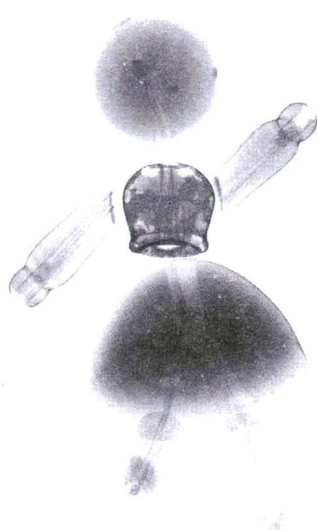
- 讲解了用 HTML 5 开发让用户最期待的游戏的经典案例。
- 展示了用 HTML 5 开发游戏创意、编程的秘诀、技巧和优化技术。

HTML 5

完美游戏开发

【白俄罗斯】Egor Kuryanovich 等 著

阳飞 海通 李碎伍 译



人民邮电出版社

北京

图书在版编目(CIP)数据

HTML5完美游戏开发 / (白俄罗斯) 库尔亚诺维奇等
著; 阳飞, 海通, 李碎伍译. -- 北京: 人民邮电出版社,
2013.6
ISBN 978-7-115-31455-0

I. ①H… II. ①库… ②阳… ③海… ④李… III. ①
超文本标记语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第066656号

版权声明

HTML5 Games Most Wanted: Build the Best HTML5 Games.

By Egor Kuryanovich, Shy Shalom, Russell Goldenberg, Mathias Paumgarten, David Strauss, Seb Lee-Delisle, Gaëtan Renaudeau, Jonas Wagner, Jonathan Bergknoff, Rob Hawkes, ISBN: 978-1-4302-3978-9

Original English language edition published by Apress Media.

Copyright ©2012 by Apress Media. Simplified Chinese-language edition copyright © 2013 by Post & Telecom Press. All rights reserved.

本书中文简体字版由 Apress L.P. 授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

内 容 提 要

本书汇集了顶尖的 HTML5 游戏开发人员的编程技术和技巧, 他们热情地与我们分享了制作和编写游戏的经验。读者将会学到很多编程的专业知识和技巧, 书中的 real-world 游戏示例还重点讲解了游戏优化的方法, 并能从中加以借鉴, 以便实际应用到自己的项目中。书中给出的技术不仅仅只是做个普通的游戏, 是要做一个超级棒的游戏。

本书囊括了 JavaScript、HTML 5、WebGL 和 CSS 3 的各种编程实例代码, 展示了如何制作那些让人乐此不疲的游戏, 以及游戏中必不可少的一些技能。不管你是一个为了让游戏一鸣惊人而寻找秘诀的专业游戏开发者, 还是一个需要动力, 想找个项目练练手积攒经验的新手, 本书都有你想要的知识和技术。书中的内容和游戏包括了由简到难的 A 到 B 游戏; 介绍了如何创建、保存和加载游戏进度的 Marble Run 游戏; 制作一个像 Cycleblob 的 3D 竞速动作游戏; 《雪人大战》(Snowball Fight)里的插值(entity interpolation)技术; 角色扮演类游戏 Grave Danger; 在 Bar Fight 游戏中学习 WebGL 的高级用法; Far7 游戏完美地结合了 Web 中的 HTML 5 技术; 做一个类似 Z-Type 游戏的秘诀。

本书适合 HTML 5 开发者、网络游戏开发者及 HTML 5 爱好者学习用书。

-
- ◆ 著 [白俄罗斯] Egor kuryanovich 等
 - 译 阳 飞 海 通 李碎伍
 - 责任编辑 张 涛
 - 责任印制 程彦红 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
 - 邮编 100061 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京艺辉印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 13.75
 - 字数: 333 千字 2013 年 6 月第 1 版
 - 印数: 1—3 000 册 2013 年 6 月北京第 1 次印刷
 - 著作权合同登记号 图字: 01-2012-8858 号
-

定价: 49.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第 0021 号

前言

用 HTML 5 开发的游戏是一场新的“游戏革命”，用它开发的的游戏、互动广告以及富媒体应用，可以在各种设备（如智能手机、平板电脑及台式电脑）的 Web 浏览器中运行，而在这以前，终端用户只能通过下载安装第三方插件（最著名的是 Flash）来实现这些功能。HTML 5 的出现让我们可以创建跨平台的 Web 应用。

本书的作者都是 real-world game 开发者，他们将与我们一起分享专业的开发经验与技巧。

第 1 章，Rob Hawkes 与我们一起讨论了当前开放式网页游戏（Open Web Game）的现状，并带领我们一起研究 HTML 5 和 JavaScript 的核心技术及全新的 API（Application Programming Interface，应用程序接口）。

第 2 章，Russell Goldenberg 会和我们一起学习他的“A 到 B”游戏的设计和开发过程。这是一个物理模拟游戏，即小球通过外界小道具（包括墙壁、加速器等）的作用，从 A 移动到 B 位置。Goldenberg 用他熟悉的 Processing.js 工具类编写了这个游戏。通过这个工具类可以简单快速地创建游戏。Goldenberg 并不是想把这个项目做成一个多人的、兼容多平台的游戏，而是在寻找一种“快速又通俗”的方法。不过，最终这个游戏成功进入了 2010 年 Mozilla 游戏大赛的决赛。

译者注：A 到 B 是一种游戏类型，就是利用游戏中提供的小道具，把某个对象移动到指定的位置。例如，爱洗澡的鳄鱼把洗澡水引入到澡盆中，又如神奇的阿力里把皮球移到小框里等。

第 3 章，Gaetan Renaudean 将向我们展示如何从零开始创建一个跨平台的 HTML 游戏。另外，她还介绍了 CSS、JavaScript、canvas 元素，以及用来编写象棋游戏的 DOM 结构。

第 4 章，将深入学习 HTML 5 中 canvas 元素的使用方法。David Strauss 和 Mathias Paumgarten 向我们展示了非常流行的 Marble Run 游戏背后的开发技术。

译者注：Marble Run 游戏网址：<http://marblerun.at/>。

第 5 章，我们会看到如何用 HTML 5 和 CSS 创建一个 3D 的 iPhone/iPad 游戏。来自荣获英国电影电视艺术学院奖的 Plug-in Media 成员 Seb Lee-Delisle 向我们展示了，如何用 CSS 3D 变换实现 HTML 元素在 3D 空间内移动。本章示例游戏中可以响应触屏的 touch 事件，达到点击游戏中的河豚鱼并把它涨破的目的。

译者注：Plug-in Media 是一家荣获英国电影电视艺术学院奖的传媒公司。他们用最先进的技术提供各种有趣的、交互性、娱乐性强的用户体验。

第 6 章的作者是 Jonas Wagnet。本章主要关注粒子系统，并利用粒子系统制作火焰、下雨和烟雾等效果。读者在本章可以学习创建一个高性能的粒子系统，处理成千上万个粒子运动。另外，还有很多示例代码，可以直接被读者借鉴使用。

第 7 章和第 8 章介绍了 WebGL。第 7 章中，Brian Danchilla 向我们介绍了 WebGL 支持检测技术、3D 坐标系统、绘制基本的图像、3D 对象动画和添加纹理等知识。

第 8 章的知识比前几章的要高级。Shy Shalom 向我们介绍了如何用 WebGL 创建三维的 CycleBlob 游戏。这个游戏原型来自电影《电子世界争霸战》（《TRON》）中光圈游戏。

译者注：游戏网址 <http://cycleblob.com/>。

第 9 章，我们跟 Jonathan Bergknoff 学习如何用 canvas、Netcode 和 WebSocket 创建一个多人在线的碰碰车游戏。这个游戏的复杂之处在于要处理多个机车的碰撞，以及为玩家添加多游戏赛道。

第 10 章中，Egor Kuryanovich 向我们介绍了如何用 HTML 5 中的技术实现“逻辑判断”（decision making）。他指出了 HTML 5 canvas 元素、SVG、WebSocket、Server-side 事件和 Web 字体的优劣，让我们充分理解了其他作者在选用这些技术时是如何进行取舍的。

源程序下载网址：www.apress.com/source-code/，或 www.apress.com/9781430239789。

本书支持社区：

9Tech 社区是新锐的综合性技术社区，以线上社区、线下活动为依托，以传播技术、资源分享为导向，集论坛、博客、下载、问答、代码库、文档库于一体，为用户提供一站式服务。目前，已有超过 32 万注册用户，超过 3000 家企业注册用户。

译林军—9Tech 社区的翻译团队，译林军成员已发展至 100 多人，均是对英语翻译和 RIA 技术兴趣浓厚的开发者们组成。译林军致力于分享国外最新技术信息，翻译国外先进的开发书籍，力求带给广大开发者最新、最有价值的技术信息。

本书适合 HTML 5 开发者、网络游戏开发者及 HTML 5 爱好者学习用书。

编者

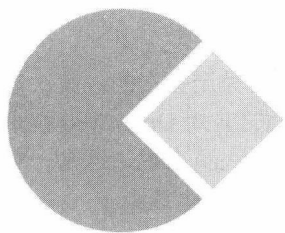
目 录

第 1 章 Open Web Game 王国..... 1	
1.1 Open Web Game 简介..... 1	
1.2 为什么要关注 Open Web Game..... 1	
1.2.1 入门很简单..... 2	
1.2.2 优秀的技术文档..... 2	
1.2.3 庞大而友好的社区..... 2	
1.2.4 一次编写, 一劳永逸..... 2	
1.2.5 代码未编译及开源..... 2	
1.2.6 全民参与..... 3	
1.2.7 拥有世界上最大的用户群..... 3	
1.2.8 Open Web Game 现状..... 3	
1.2.9 现有游戏浏览器技术..... 4	
1.2.10 canvas 和 SVG 实现 2D 图形..... 4	
1.2.11 WebGL 实现 3D 图形..... 4	
1.2.12 requestAnimationFrame 实现高性能动画..... 4	
1.2.13 HTML5 音频和音频 数据处理 API..... 5	
1.2.14 WebSocket 实现多人 实时游戏..... 5	
1.2.15 数据本地存储 IndexedDB/ Local Storage 和 其他 API..... 5	
1.2.16 通过应用缓存离线 玩游戏..... 6	
1.2.17 全屏 API 让游戏 身临其境..... 6	
1.2.18 鼠标锁定 API 让鼠标 变听话..... 6	
1.2.19 GamePad API 实现 console 般游戏体验..... 7	
1.2.20 BrowserID 服务识别 用户..... 7	
1.2.21 创建 WebRT 一样的本地 OS 应用..... 7	
1.2.22 未来的路还很长..... 8	
1.2.23 现有的出色的 Open Web Game..... 8	
1.2.24 现有的游戏发布和 盈利方式..... 10	
1.2.25 Open Web Game 的缺点..... 11	
1.2.26 未来发展方向..... 12	
第 2 章 化繁为简..... 13	
2.1 自我介绍..... 13	
2.2 A 到 B 游戏..... 14	
2.2.1 影响和灵感..... 15	
2.2.2 游戏创作阶段..... 15	
2.2.3 准备阶段..... 16	
2.2.4 正式开发阶段..... 16	
2.2.5 游戏发布..... 17	
2.2.6 积分榜..... 18	
2.3 简化原则..... 18	
2.4 A 到 B 游戏何以成功..... 20	
2.5 Processing.js..... 20	
2.5.1 Processing.js 如何工作..... 21	
2.5.2 如何使用 Processing.js..... 21	
2.5.3 Processing.js 的优缺点..... 21	
2.5.4 是否适合开发游戏..... 22	
2.6 Processing.js 实例和整合..... 22	
2.6.1 Processing 基本语法..... 22	
2.6.2 嵌入 sketch..... 22	
2.6.3 嵌入 Processing 代码..... 23	
2.6.4 整合 JS..... 24	
2.6.5 使用音频..... 24	
2.6.6 导入和加载图像..... 25	
2.6.7 jQuery 实现积分榜..... 25	
2.7 小结..... 26	

2.8	资源分享	26	4.5	HTML 标签	55
第 3 章	从零开始制作多平台 HTML5 游戏	27	4.6	加一点点样式	57
3.1	简介	27	4.7	JavaScript 结构	58
3.2	三部分代码结构	28	4.8	循环绘制	60
3.2.1	HTML 标记语言	29	4.9	砖块	64
3.2.2	CSS 样式描述语言	31	4.9.1	矩形砖块	64
3.2.3	制作页面缓动效果	35	4.9.2	圆形砖块	65
3.2.4	JavaScript 高级动态语言	37	4.9.3	扇形砖块	66
3.3	编写象棋游戏	38	4.9.4	三角形砖块	67
3.3.1	游戏规则	38	4.10	将砖块添加到网格中	68
3.3.2	游戏样式	39	4.11	角度	71
3.3.3	游戏逻辑	40	4.12	清除按钮	72
3.4	移动设备框架	44	4.13	存储	73
3.4.1	jQuery 和 Zepto	44	4.13.1	保存路径	73
3.4.2	PhoneGap、类库	44	4.13.2	还原路径	74
3.4.3	PhoneGap Build	44	4.14	保存按钮	75
3.4.4	象棋游戏的配置	45	4.15	还原保存的路径	76
3.4.5	WebAppBuilder	45	4.16	总结	77
3.4.6	象棋游戏的 Makefile	46	第 5 章	详解 3D CSS	78
3.5	国际化	47	5.1	3D CSS 简介	78
3.6	纯 DOM 游戏 vs canvas 游戏	49	5.2	HTML 元素总览	78
3.6.1	canvas API 回顾	49	5.3	游戏变量	79
3.6.2	方案对比: canvas	49	5.4	Container 的 3D 属性	79
3.6.3	方案对比: 纯 DOM	50	5.5	事件和游戏计时器	80
3.6.4	canvas 和 DOM 混合使用	50	5.6	游戏循环总览	80
3.7	小结	50	5.7	创建新的小鱼	81
3.8	其他资源	51	5.8	更新小鱼	83
第 4 章	创建、保存和加载路径	52	5.9	渲染小鱼	84
4.1	我们是谁? 要做什么?		5.10	创建小鱼	85
	为什么要做	52	5.11	设置小鱼属性	85
4.2	应用及意义	53	5.12	回收小鱼	86
4.3	为何关卡很重要	53	5.13	小鱼爆炸	86
4.4	化整为零	54	5.14	粒子	87
			5.15	平行视差图层	88

5.16	禁用默认的触控/鼠标行为	88	6.6	优化	106
5.17	iPad 上的河豚鱼游戏	88	6.6.1	单个粒子	106
5.18	结论	89	6.6.2	类型化数组	107
第 6 章	粒子系统	90	6.6.3	requestAnimationFrame	108
6.1	粒子系统简介	90	6.7	细节优化	108
6.2	数学知识	90	6.7.1	链式表达式	108
6.2.1	向量	90	6.7.2	取舍操作	108
6.2.2	增加检测	92	6.8	优化焰火	109
6.2.3	随机值	93	6.9	总结	111
6.2.4	粒子系统的组件	93	第 7 章	开始 WebGL 之旅	112
6.2.5	粒子系统中的粒子	93	7.1	WebGL 起源	112
6.2.6	发射器	94	7.2	WebGL 是如何工作的	113
6.2.7	粒子系统的作用力	94	7.2.1	用途和局限	113
6.2.8	渲染器	95	7.2.2	演示	113
6.2.9	系统	95	7.3	操作系统, 显卡, 浏览器支持	113
6.3	你好焰火	96	7.4	库	115
6.3.1	设计火花	96	7.4.1	矩阵库	115
6.3.2	实现主循环	97	7.4.2	高级 API	115
6.3.3	实现发射器	97	7.5	调试工具	116
6.3.4	作用力	98	7.6	3D 刷新器	116
6.3.5	生存和死亡	99	7.6.1	坐标系统	116
6.3.6	渲染	100	7.6.2	变换	116
6.4	火	100	7.6.3	组合场景视图	116
6.4.1	创建纹理	100	7.7	GLSL	118
6.4.2	装载纹理	101	7.8	使用 WebGL	119
6.4.3	实现发射器	101	7.8.1	画一个简单的形状	119
6.4.4	作用力的实现	102	7.8.2	动画和景深 (depth)	129
6.4.5	实现主循环	102	7.9	一个基础的游戏	136
6.4.6	旋转	103	7.10	总结	147
6.4.7	实体的渐隐	103	第 8 章	CycleBlob: 一个基于 WebGL 的 Lightcycle 游戏	148
6.5	烟	104	8.1	什么是 WebGL	148
6.5.1	创建纹理	104			
6.5.2	实现发射器	105			
6.5.3	实现主循环	105			

8.2	开始 WebGL 之旅	149	9.7	结论	195
8.3	3D 模型和几何形状	151	9.8	附录: 设置 node.js	195
8.4	移动和定位	158	9.8.1	Windows	195
8.4.1	网格之上	158	9.8.2	UNIX	196
8.4.2	摄像机定位	161	第 10 章	艰难技术的选择	197
8.4.3	动画引擎	163	10.1	为什么选择很难做	197
8.5	代码结构	169	10.2	渲染	198
8.6	结论	173	10.2.1	SVG	198
8.7	参考	174	10.2.2	canvas	200
第 9 章	使用 WebSocks 实现实时多人游戏	175	10.3	音频	203
9.1	网络代码的哲学	175	10.4	连接	204
9.2	设计碰碰车游戏	176	10.4.1	服务器端发送事件	205
9.3	游戏逻辑	177	10.4.2	WebSocket	207
9.4	游戏客户端, 第一部分	181	10.5	客户端数据库	208
9.5	游戏服务器	184	10.6	页面字体	209
9.6	游戏客户端, 第二部分	191	10.7	结论	210
			索引		211



第1章 Open Web Game 王国

本章将介绍通过 HTML5 和 JavaScript 等技术开发 Open Web Game 的理念, 包括这些技术带来的一些新特性, 如 GamePad API 可以摆脱键盘和鼠标输入。在本章结束部分, 介绍了如何发布游戏, 以及如何让游戏给你带来盈利。另外, 本章还列举了一些在 Web 发展成为游戏开发平台可行性方案的过程中, 必定要发生的几个事件。

1.1 Open Web Game 简介

我叫 Rob Hawkes, 是一个串行实验者(serial experimenter), 专注于 JavaScript 开发。我是 Mozilla 的一名技术布道者, 对 Mozilla 以及广大开发者社区发布的游戏、相关事件和项目有着浓厚的兴趣。

我还是一个业余的游戏开发者, 曾经用各种不同的语言(Unity、ActionScript、PHP、Processing 和 JavaScript)及不同的技术开发过各种不同的项目。

最近, 我正在开发一个名为 Rawkets (<http://www.rawkets.com/>) 的 HTML5 多人游戏, 在这个游戏中, 我测试了各种游戏相关的技术, 稍后我将向你一一道来。

任何问题都可以通过我的个人博客 (<http://www.rawkes.com/>) 或者 twitter (@robhawkes) 与我联系, 我会尽力帮助你。

Mozilla 基金会简称“Mozilla”(缩写“MF”或“MoFo”), 是为支持和领导开源的 Mozilla 项目而设立的一个非营利组织。该组织制定管理开发政策, 经营关键基础组织并管理商标及其他知识产权。

1.2 为什么要关注 Open Web Game

Open Web Game 是基于 Open Web 技术开发的原生游戏, 而 Open Web 技术是指 HTML5、CSS 和 JavaScript。Open Web Game 凝结了这些技术的主要特性, 所以与其一个个学习这些技术, 我更愿意在 Open Web Game 这个庞大的题目中学习, 因为可以综合这些技术练习。因此, 我们要对 Open Web Game 心存感激。

Open Web Game 开发技术魅力在于, 它跟网站或网页应用开发(有人说这两个定义是相同的, 不过, 这里我还是把它们分开了)用到的技术是一样的。在 Internet 诞生的时候, 这些技术就已经非常盛行了, 而且事实证明, 在各种技术此起彼伏的时候, 它们是最可靠、最稳定的。

1.2.1 入门很简单

用 Open Web 技术开发游戏几乎没有任何障碍。首先, Open Web 技术是完全免费的;其次, Open Web Game 开发所需要的工具都有免费版或者只需极少的费用。简而言之,用 Open Web 技术开发游戏只需极少的投入。对于那些用指定开发环境(如 Flash IDE)的开发人员来说,这是一个很大的亮点,因为开发者通常需要向这些开发环境购买一些技术支持和相关的开发工具。

1.2.2 优秀的技术文档

除了“零障碍”入门,良好的编写风格和免费的技术文档,都可以帮助开发者学习这些技术的每一个细节。像 Mozilla Developer Network 网站在几年前就已经开始提供这种服务了。另外,还有大量其他的网站和个人博客也会发布相关的信息。

1.2.3 庞大而友好的社区

有这样一个流行的社区,它由一群持续关注 Web 的发展趋势、分享经验的开发人员和设计人员组成,他们竭力为大家提供更多的精品技术文章。这样的社区随处可见:从 Facebook 的 HTML5 小组,到 Tiwtter 的 HTML5 小组,到专门的技术论坛(如 SitePoint),再到 IRC 上实时聊天室。另外,现在 irc.freenode.net 上新开了一个 Open Web Game 频道,期待你的加入。

1.2.4 一次编写,一劳永逸

从更专业的角度看,Open Web 技术开发游戏的魅力在于,它是一个“一次编写,一劳永逸”的技术。现在可能还不完全如此,因为各个应用平台上的使用规范还是有些细微的差异。但 Open Web 是一个原生的跨平台技术,除了少许的非指定平台专用代码外,已经在各种不同的操作系统和浏览器中应用。

可以肯定的一点,如果一个平台支持游戏所需的 HTML5 和 JavaScript API,那么游戏运行状况应该可以跟开发者预想的一样。当然不同的硬件配置对性能会有不同的影响,不过这也是其他技术同样无法避免的问题。

1.2.5 代码未编译及开源

一些拥有个人专利的开发者在转向 Open Web 技术时,会感觉很不习惯,因为代码没有经过任何编译,开源的给所有用户查看。在浏览器的页面中,单击鼠标右键,可以轻松地查看页面的源代码和素材——Open Web Game 亦是如此。

这种现象同时也是 Open Web 概念和强大的核心所在,无法撼动。但是很多从其他平台转行过来的开发人员试图改变这个特性。数字版权加密保护技术(digital rights Management, DRM)试图以一种非专利方式管理 Web,也可以说是对代码和素材进行编译。这样其他人就无法那么轻易地获取这些资源了,不过这未必能实现。

译者注

由于数字化信息的特点决定了必须有另一种独特的技术,来加强保护这些数字化的音视频节目内容的版权,该技术就是数字版权加密保护技术 DRM。

在我看来,这个文化上的冲突,是其他平台(如手机、Console 和 PC)上转行过来的开发者一直担忧的主要问题。传统的游戏开发理念是保护知识产权,使代码和素材无法被获取。有人尝试过在 Open Web 中隐藏代码和素材,但最终毫无收获(我尚未看到 DRM 提出一个真正的可行的方案)。我能理解他们的想法以及他们的目的。不幸的是,没有浏览器会支持他们的观点,所以对 Open Web Game 不适合。

译者注

Console 是指游戏主机,如 Sony 的 PSP 及微软的 Xbox 等都叫做 Console。

向开发者证明 Web 是个可行的开发平台,并证明它对开放资源和代码可以进行保护,不至于被疯狂盗用,失去控制,是很重要的。我坚信,历史告诉我们盗用或复制都不会发生,我坚信,Web 作为游戏开发平台所带来的利益会远远大于这些细小的不足。换句话说,你见过多少盗用或复制的网站,会比原始网站要好?

1.2.6 全民参与

还有一点让我很吃惊,即没有任何一个组织来管理 Web 开发技术和平台。这对从专利背景转过来的开发者来说是个很陌生的概念,因为当他们想要添加或变更什么东西时,找不到对应的窗口。

Web 开发技术是由一系列规范来定义的。这些规范分别由万维网(World Wide Web, W3C)或者网页超文本应用技术工作组(Web Hypertext Application Technology Working Group, WHATWG)来管理。这两个组织由各种不同的股东组成,从浏览器开发商到科技技术公司,再到普通的 Web 开发者,任何人都可以成为组织中的一员。这就是为什么说全民参与了。如果需要添加或变更说明什么,只需要加入讨论,就能畅所欲言。如果你想参与讨论 Open Web Game 技术,请加入 W3C 游戏社区组织(www.w3.org/community/games/)。

1.2.7 拥有世界上最大的用户群

Web 开发游戏的优点还在于 Web 拥有世界上最大的用户群。通过 Web 技术,可以接触到 69 亿的 Web 用户。当然我承认并不是每个人都会把浏览器升级到最新,所以不可能接触到所有的用户。但我想说的是,Web 用户的数量是非常惊人的,而且每天都有越来越多的用户连接到 Web 上。即便只计算 Firefox 火狐浏览器的用户,也有上亿(好多啊)。Facebook 用户呢?有超过 8 亿的用户。

如何发布游戏给广大的用户玩,是另外一个问题。我会在本章的后面章节中进行简单地说明。

1.2.8 Open Web Game 现状

在过去的几年中,浏览器平台和 Open Web Game 开发技术取得了突飞猛进的进步。这与 APP 商店和社交网络中不断增涨的 Open Web Game 是分不开的,而且这个数量每天都在增加。此外,一些大型的游戏公司也开始对 Open Web Game 产生兴趣,使得游戏的质量也呈明显改善趋势。但是,我觉得在这些改善中最重要的一点是 Flash 将不再支持移动设备和 Web,而 HTML5 和 JavaScript 是一个永远不会因为兼容性问题被抛弃的平台——Open Web Game 应运而生。

1.2.9 现有游戏浏览器技术

我觉得 Web 领域中最吸引人的除了现有的技术,还有即将到来的新兴技术,比如 GamePad API、鼠标锁定 API 和全屏 API 等。这些简易的技术会帮助我们摒弃“Web 游戏只不过是网页中嵌入的小盒子”的观念。有了连接游戏机控制器,就有了全屏运行 HTML 元素的能力,Open Web Game 也会变得像 Console 和 PC 一样,带来更诱人的用户体验。

接下来,我举例说明一些现有的和即将到来的新兴技术,建议你好好看看,从中你会发现入门其实很简单。值得一提的是,在这些技术更新时,浏览器会相应地很快做出更新来支持这些技术。我经常去 When Can I Use 网站关注这些更新信息 (<http://caniuse.com>)。

1.2.10 canvas 和 SVG 实现 2D 图形

因为图形输出是大多数游戏的核心组件之一,所以创建和管理 2D 图形的能力对浏览器来说很重要。这样一来,HTML5 canvas 和可变矢量图形 (Scalable Vector Graphics, SVG) 就应运而生了。

译者注

可缩放矢量图形是基于可扩展标记语言 (XML),用于描述二维矢量图形的一种图形格式。SVG 由 W3C 制定,是一个开放标准。另外,SVG 还是 Static Var Generator (静止无功发生器) 的简称。

HTML5 canvas (通常简称“canvas”)是一个 JavaScript API,与同名的 HTML 元素相对应,可以在浏览器中创建和编辑位图图形。它的优点是快速,且创建像素级图像而不影响运行速度。它的缺点是性能在不同的平台上不尽相同,并且没有内置的动画功能。

另外一个 2D 解决方案是 SVG,使用文件物体模型 (Document Object Model, DOM) 来创建和管理矢量图形。它的优点是易于读取 (图形都用 DOM 元素表示),并且有内置的动画功能,另外因为是矢量图形,所以图形可以轻松缩放以适应不同的屏幕大小。它的缺点是没有像 canvas 那样流行,对完美的像素级操作处理得不是很好。

1.2.11 WebGL 实现 3D 图形

如果需要在游戏中添加一些 3D 图形,那就需要使用到 WebGL JavaScript API (简称 WebGL)。WebGL 基于 OpenGL ES2.0,包含了一些炫酷效果所需的所有功能。

WebGL 的优点是基于硬件加速 (非常快),可以创建非常复杂的视觉效果。它的缺点是,学习起来非常复杂,而且不支持 IE 浏览器。不过通过第三方框架,比如 three.js,可以大大降低学习的难度。

1.2.12 requestAnimationFrame 实现高性能动画

大部分的 Open Web Game 中的动画都是通过我们常说的循环,重复不断地更新屏幕实现显示的。当更新速度足够快时,图像的更新就会非常平滑,形成动画。

到目前为止,实现动画最简单的方法还是用 JavaScript 的 setTimeout 和 setInterval。需要注意的是,这两种方法会使代码持续运行,带来一系列的性能问题。另外,当游戏页面处在非活动状态

或浏览器最小化的时候，这两个方法不会停止运行，使得动画实现起来很不理想。

为了解决这个问题，HTML5 引入了 `requestAnimationFrame` 方法，目的是将动画循环管理功能指派给浏览器。这样就尽可能以最佳的方式来实现动画，同时大幅度提升游戏性能，防止在非活动页面或者最小化浏览器中持续运行动画。通过这个方法可以大幅度降低动画循环的速度，甚至是停止播放。这对延长手机电池使用寿命等有非常好的效果。

1.2.13 HTML5 音频和音频数据处理 API

对于大部分游戏来说，另外一个基本的元素是音频。截止到目前，在 Web 中实现某些音频效果还存在困难，因此很多时候都选择使用 Flash。HTML5 audio 元素的到来为我们解决了这个问题。它可以在浏览器中无插件播放音频。

HTML5 audio 元素还是存在一些局限性的，比如它的目的只是播放单个音频文件，如游戏中的背景音乐，而不适合做音频效果，特别是那些快节奏、一次播放多个音频文件的效果。

为了解决这个问题，Mozilla 和 Chrome 分别引入了 Audio Data API 和 Web Audio API，提供了更多详细的音频功能。通过这些 API，我们可以用 JavaScript 创建声音，可在运行时编辑音频，同时播放多个音频通道，还可以获取音频在实时播放时的数据。

遗憾的是，音频数据 API 解决方案还没有一个统一的标准。因此，在使用时，对于 Mozilla 和 Chrome 内核的浏览器要做些细微的调整。我希望在不久的将来能有一个统一的标准规范音频数据 API。

1.2.14 WebSocket 实现多人实时游戏

提到多人游戏，你想到的可能是实时发送 Ajax 请求（带来无法忍受的延迟）及 Flash。但这两种方法都不是很理想。不过那个年代将一去不复返了。从 2011 年起，WebSocket 进入了各大浏览器（是的，包括 IE10），实现了浏览器与服务器之间的双向实时通信。

为什么双向实时通信对游戏那么重要呢？实时通信从字面上来说，就是数据可以在用户浏览器中实时地流入流出。它的一个显著的优势是节省带宽。不像 Ajax 需要定时发送需求来检查新的数据。WebSocket 会始终保持连接，在有需求时，立即将数据添加到客户端和服务端。这个优势在每几毫秒更新一次的快节奏游戏中展现得淋漓尽致。除此之外，WebSocket 的双向通信特性，也表示数据可以从服务器端到客户端，以及从客户端到服务器端即时传送。

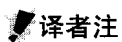
1.2.15 数据本地存储 IndexedDB/ Local Storage 和其他 API

很多游戏都可以将玩家数据保存到玩家的电脑上，以便以后可以读取这些数据继续游戏（比如游戏存档或图形缓存等）。过去，唯一的方法是将数据保存到服务器上，要忍受重新加载的延时，或者使用 cookies 之类的方法，且只能保存很少的数据。

幸运的是，现在已经有很多不同的方法来解决这个问题。最常用的是 IndexedDB、LocalStorage 和各种 File 和 FileSystem API。前两个方法可以在用户浏览器中存储大量结构化数据，Indexed DB 方法甚至还可以存储文件。文件（File）和文件系统（FileSystem）API 使游戏可以通过 JS 读取用户操作系统中的文件，可以保存和读取更大的数据。这一点是其他任何一种方法都无法比拟的。

1.2.16 通过应用缓存离线玩游戏

在 Web 中创建游戏的想法是不错的，但是如果玩家想在离线状态玩游戏呢？或者说，当玩家正在玩 EPIC 的游戏时，网络突然掉线了，会怎么样呢？从目前的 Open Web Game 来看，最坏的情况是掉线后游戏立即停止。最好的情况是向服务器端发送数据，保存完当前的游戏数据。在未发现新技术时，掉线的玩家刷新页面后，游戏会继续运行，不过玩家们看到的将会是一个空白的页面，而且他们在离线状态下辛辛苦苦玩的游戏进度全部丢失。玩家对此会很不爽。让玩家不爽是很糟糕的。



译者注

EPIC 指 EPIC 公司 (Epic Games)。Epic Games 是全球领先的数字游戏及图形交互技术开发商，1991 年创立于美国。著名电脑游戏虚幻 (Unreal) 和《虚幻竞技场 (Unreal Tournament)》系列是 Epic Games 的成名作。

现在已经有一些方法可以帮我们解决这个问题了。首先是使用缓存。通过缓存文件保存一些特定的元素（如 HTML、CSS、图片和 JavaScript），即一些你希望浏览器缓存下来供离线使用的元素。不仅如此，通过这些缓存文件，可以加快游戏的加载过程。

另外一个方法是将玩家游戏数据存储到本地电脑中，然后周期性地与服务器同步。过去浏览器无法存储足够的数据来实现这个功能，但是当 Local Storage 和 IndexedDB 出现后，我们可以有条不紊地存储上百兆的数据。

另外还可以在游戏中的添加一个提示功能，当网络断开时提示玩家。JavaScript 中的 navigator.online 属性可以用来检测用户网络是否处在连线状态。还可以用 offline 和 online 事件，在网络连接发生变化时，自动触发游戏的一些行为，比如停止所有的 WebSocket 通信，将用户数据缓存在本地电脑中，直到网络连接恢复。

1.2.17 全屏 API 让游戏身临其境

现在的网页游戏看起来都像是嵌入在页面中的小盒子，这使游戏很难让人感觉到身临其境。为什么会这样呢？因为它实际上就是嵌入在页面中的小盒子。午饭时间花个五六分钟，玩玩简单的益智游戏，虽然是在小盒子里，周围是浏览器 UI 和其他让人分心的东西，但感觉也还好。不过，如果是玩第一角色的射击游戏或赛车游戏，那感觉就不怎么样了。

很幸运的是，现在我有 FullScreen 全屏 API 来解决这个问题。它可以将任何一个 DOM 元素填满用户的整个屏幕，而不仅仅是视频。它可以带来与几分钟的益智游戏完全不同的游戏体验，让你身临其境地玩上几个小时。

1.2.18 鼠标锁定 API 让鼠标变听话

在玩家用鼠标围着一个固定点旋转（比如 3D 第一角色射击游戏）或者在 2D 俯视游戏中用鼠标控制方向时，经常会遇到的问题是鼠标不听使唤，且鼠标一直都是可见的，这时会很烦人，会毁掉游戏体验。让游戏体验变得更糟糕的是，鼠标离开浏览器窗口时，所有的运动都会停止。在全屏模式下，当用户的鼠标移动至屏幕边缘时，也会出现同样的问题。对于用户来说，这个问题很可恶，会完完全全毁掉游戏体验。

好消息是, Mouse Lock 鼠标锁定 API 已经诞生了。它专门用来解决这个问题。不过, 现在只有试行版的 Firefox Nightly 支持这个 API, 不久它会登录 Chrome (在读到此文时, 这两个浏览器的正式版应该都已经支持了这个 API)。这个 API 唯一的目的是隐藏鼠标, 并将其锁定在一个固定的位置, 使它无法到达屏幕边缘。就是说, 以前以浏览器左上角为基准点的坐标系, 将被替代为以鼠标与锁定点相对距离决定。

1.2.19 GamePad API 实现 console 般游戏体验

Web 游戏中另外一个与输入相关的改善当属 GamePad API 了。对于玩家来说, 鼠标和键盘将不再是游戏过程中唯一的输入方案。有了 GamePad 我们可以通过 JavaScript 调用各种游戏手柄, 甚至包括一些 console 的控制器, 比如 Xbox360 和 PS3 (但是要有第三方驱动)!

 译者注

微软 Xbox 360 是唯一一款具备定时功能的游戏机, 家长们可轻松设定相应游戏时间, 同时也能对孩子们所玩、所观看的内容加以限制。PlayStation 3 简称 PS3, 是索尼电脑娱乐所开发的家用游戏机, 也是该公司推出的第三款电视游戏机。

和 Mouse Lock API 一样, 目前只有试行版的 FireFox Nightly 和 Chrome 支持 GamePad API, 而且用起来非常简单好用(同样, 在你读到本文时, 正式版的 Firefox 和 Chrome 应该已经支持 GamePad API 了)。结合着 FullScreen API, GamePad API 可以让 Open Web Game 的游戏体验从网页里面的小游戏完全蜕变为桌面游戏和 Console 游戏。

1.2.20 BrowserID 服务识别用户

正如 iOS 提供的 OpenFeint 和 Apple Game Center 服务, Web 游戏需要有一个公开可靠的方法来识别用户。BrowserID 是 Mozilla 对这个问题的解决方法。它允许用户用自己的电子邮箱登录游戏而不需要密码。

 译者注

OpenFeint 是一个基于 SNS 的平台, 为 iOS 和 andriod 系统提供在线游戏竞技的技术平台游戏中有很多与用户相关的功能, 比如好友列表、积分榜、聊天或多人游戏功能, 第一步要做的是识别用户。

1.2.21 创建 WebRT 一样的本地 OS 应用

Mozilla 更为显著的一个特性是集成了 Web 实时运行环境 (Web run-time, WebRT), 这是首创。用户可以将游戏安装到自己的本地操作系统 (Windows、Mac 和现在流行的 Android) 中, 然后向其他 OS 一样, 有一个发布后的应用图标。

WebRT 是以应用为中心的用户代理 (app-centric user agent) 形式运行游戏的, 与 Firefox 中以浏览器为中心的用户代理 (browser-centric user agent) 不同, 其使用的是单独的用户配置和系统进程运行。

WebRT 分出单独的进程运行游戏, 同时移除浏览器的 UI (User Interface, 用户界面), 使游戏

用户体验更加有亲和力。这与点击 Mac 的 dock 工具中的图标在本地窗口中运行游戏有异曲同工之妙。用户丝毫没有在浏览器中玩游戏的感觉。

对于一个开发者来说,这一切看起来有点虚幻。WebRT 使 Web 游戏不再只是一个漂亮的页面,而是一个真正的应用,即拥有自己的用户体验的应用。用我的话来讲,这将是 Web 游戏从 5 分钟的益智游戏到拥有 Console 游戏版用户体验的专业级游戏转变过程的转折点、里程碑。

1.2.22 未来的路还很长

前面所提到的技术只是真正的 Open Web Game 时代中的 Open Web 技术的冰山一角。Mozilla 和其他的厂商也在努力为我们带来更多这样的 API 和服务,帮我们把 Web 打造成更好的游戏环境。

1.2.23 现有的出色的 Open Web Game

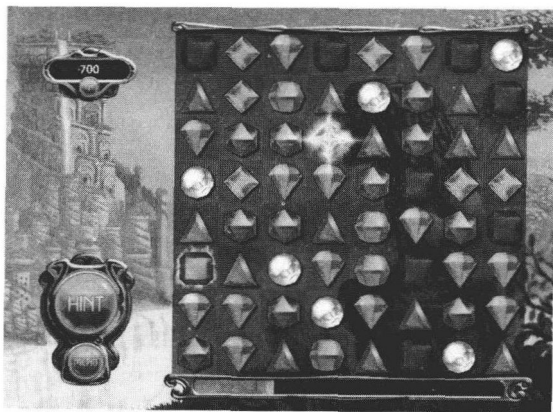
尽管 Open Web Game 开发是个相对新兴的技术,但是现在已经创造了很多相当出色的游戏。下面简单地列举其中的几个。

1. 宝石迷阵

大约在 2011 年年底,PopCap 发布了非常火爆的“宝石迷阵”(Bejeweled)游戏的 HTML5 版(如图 1-1 所示)。它使用 WebGL 技术进行图像加速。如果不支持 WebGL 的话,会用 HTML5 canvas 取代。用户可以登录 <http://bejeweled.popcap.com/html5/0.9.12.9490/html5/Bejeweled.html> 来试玩这个游戏。

译者注

Popcap Games 是世界领先的休闲游戏的开发商和发行商,在 2000 年由 John Vechey、Brian Fiete 和 Jason Kapalka 共同建立,总部位于美国的西雅图。



▲图 1-1 HTML5 版宝石迷阵截图

2. 愤怒的小鸟

愤怒的小鸟可以说是现在最流行的游戏了。在今年年初的时候,Revio 公司发布了 HTML5 版