

<<HTML5游戏开发实战>>

图书基本信息

书名：<<HTML5游戏开发实战>>

13位ISBN编号：9787111391760

10位ISBN编号：7111391764

出版时间：2012-7

出版时间：机械工业出版社

作者：Makzan

页数：260

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<HTML5游戏开发实战>>

前言

HTML5有望成为网络游戏开发的热门新平台。

HTML5游戏能够运行于包括iPhone系列和iPad系列在内的计算机、智能手机以及平板电脑上。

今天，我们加入HTML5游戏开发的急先锋，明日将成为时代的弄潮儿。

本书将展示如何使用最新的HTML5和CSS3 Web标准来构建纸牌游戏、绘图游戏、物理游戏，甚至是多人网络游戏。

本书讲解清晰、系统全面，参照本书的讲解就能构建出6款示例游戏。

HTML5、CSS3以及相关的JavaScript API是网络上最新的热点话题。

这些标准给我们带来一个新的游戏市场——HTML5游戏。

通过新的功能，我们能用HTML5元素、CSS3属性，以及JavaScript来设计运行于浏览器的游戏。

本书根据关注话题的侧重点而分成9章。

我们将通过创建6款游戏来具体学习如何绘制游戏对象、绘制动画、加入音频、连接玩家，以及用Box2D物理引擎构建游戏。

本书内容：第1章介绍HTML5、CSS3，以及相关的JavaScript API 新功能。

该章还演示了利用这些功能和特性能创建什么样的游戏。

第2章通过使用DOM和jQuery创建传统《乒乓球》游戏来开始我们的游戏开发之旅。

第3章探讨CSS3新功能，讨论如何用DOM和CSS3来创建《纸牌记忆配对》游戏。

第4章介绍在页面中通过新Canvas元素来绘制游戏并与之进行交互的新方式，还演示如何用Canvas来构建《解题》游戏。

第5章通过在Canvas中绘制渐变效果和加入图像进行美化，完善前面介绍的《解题》游戏。

该章还讨论基于帧的动画精灵和多层管理方法。

第6章使用Audio元素给游戏添加声音效果和背景音乐。

该章还讨论各浏览器对声音格式的兼容性，并在该章结尾完成一款由键盘驱动的音乐游戏。

第7章使用新的本地存储API保存和恢复游戏进度，记录最佳成绩功能，进一步增强第3章所创建的CSS3《纸牌记忆配对》游戏。

第8章讨论能让浏览器持久连接到socket服务器的新WebSocket API，用于实现多人联网玩游戏，并在该章完成《我画你猜》游戏。

第9章传授如何把Box2D(著名物理引擎)集成到Canvas游戏中。

该章讨论如何创建物体、应用力、连接物体，以及如何把图形与物体进行关联，并最终创建一款平台汽车游戏。

阅读本书你需要做的准备 你需要最新的现代Web浏览器，一个优秀的文本编辑器，HTML、CSS以及JavaScript的基本知识。

本书读者对象 本书适合于对HTML、CSS和JavaScript有基本理解，并希望设计网页类Canvas游戏、基于DOM游戏的设计者。

<<HTML5游戏开发实战>>

内容概要

这是一本经典的HTML5游戏开发教程，它清晰而全面地展示了如何使用最新的HTML5和CSS3标准来构建各种常见类型的游戏。

本书实战性非常强，围绕纸牌游戏、绘图游戏、物理游戏和多人游戏精心组织了6个富有趣味性和技术性的游戏案例。

如果动手实践这些案例，不仅能掌握各种类型游戏的开发思路和设计方法，而且还能掌握HTML5和CSS3技术中与游戏开发相关的理论知识。

本书共分9章，第1章介绍HTML5、CSS3，以及相关的JavaScript API

新功能；第2章通过使用DOM和jQuery创建传统游戏；第3章讨论如何用DOM和CSS3来创建游戏；第4章介绍在页面如何通过Canvas来绘制游戏并与之交互；第5章介绍如何在Canvas中绘制渐变效果和加入图像；第6章使用Audio元素给游戏添加声音效果和背景音乐；第7章使用新的本地存储API保存和恢复游戏进度，进一步增强游戏；第8章讨论如何实现多人联网玩游戏；第9章讲解如何把Box2D集成到Canvas游戏中。

本书以实例为导向，系统介绍网络游戏开发技术，结合具体示例的操作步骤讲解，浅显易懂，适合网络游戏开发人员、管理人员阅读。

<<HTML5游戏开发实战>>

书籍目录

审校者简介

译者序

前言

第1章 HTML5游戏概述

1.1 探索HTML5新功能

1.1.1 Canvas

1.1.2 音频

1.1.3 GeoLocation

1.1.4 WebGL

1.1.5 WebSocket

1.1.6 本地存储

1.1.7 离线应用程序

1.2 探索CSS3新功能

1.2.1 CSS3转换

1.2.2 CSS3变换

1.2.3 CSS3动画

1.3 HTML5和CSS3新功能的更多细节

1.4 创建HTML5游戏的好处

1.4.1 不需要第三方插件

1.4.2 不需要插件就能支持iOS设备

1.4.3 突破常规浏览器游戏限制

1.4.4 创建HTML5游戏

1.5 HTML5还能做些什么

1.5.1 《记忆配对》游戏

1.5.2 Sinuous

1.5.3 小行星式书签小程序

1.5.4 Quake

1.5.5 蝌蚪聊天室

1.5.6 Scrabb.ly

1.5.7 Aves引擎

1.6 浏览更多HTML5游戏

1.7 本书主要涉及哪些游戏

1.8 总结

第2章 DOM游戏开发入门

2.1 准备开发工具

2.2 为DOM游戏准备HTML文档

2.2.1 新式HTML5 doctype声明

2.2.2 页眉和页脚

2.2.3 JavaScript代码最佳位置

2.2.4 页面加载完后运行代码

2.3 设置《乒乓球》游戏元素

2.3.1 jQuery简述

2.3.2 jQuery选择器基础知识

2.3.3 jQuery css函数

2.3.4 使用jQuery的好处

<<HTML5游戏开发实战>>

- 2.3.5 使用jQuery操纵DOM游戏元素
- 2.3.6 绝对位置的行为
- 2.4 获取玩家的键盘输入
 - 2.4.1 了解按键代码
 - 2.4.2 让常数更具可读性
 - 2.4.3 用parseInt函数将字符串转换为数字
 - 2.4.4 在控制台面板中直接执行JavaScript表达式
 - 2.4.5 检查控制台窗口
- 2.5 支持玩家多键盘同时输入
 - 2.5.1 更好地声明全局变量
 - 2.5.2 用setInterval函数创建JavaScript定时器
 - 2.5.3 理解游戏主循环
- 2.6 通过JavaScript间隔移动DOM对象
- 2.7 开始碰撞检测
- 2.8 动态显示HTML文本
- 2.9 总结
- 第3章 用CSS3构建《纸牌记忆配对》游戏
 - 3.1 用CSS3转换和变换模块移动游戏对象
 - 3.1.1 2D变换函数
 - 3.1.2 3D变换函数
 - 3.1.3 用CSS3转换实现样式间过渡
 - 3.2 创建翻牌效果
 - 3.2.1 使用jQuery toggleClass函数切换类
 - 3.2.2 通过z-index控制重叠元素的可见性
 - 3.2.3 介绍CSS perspective属性
 - 3.2.4 介绍背面可见性
 - 3.3 创建《纸牌记忆配对》游戏
 - 3.3.1 下载纸牌精灵表图像
 - 3.3.2 设置游戏开发环境
 - 3.3.3 使用jQuery复制DOM元素
 - 3.3.4 使用jQuery的子节点过滤器选择首个子元素
 - 3.3.5 垂直对齐DOM元素
 - 3.3.6 通过背景位置来使用CSS精灵
 - 3.4 给配对游戏添加游戏逻辑
 - 3.4.1 在CSS转换完后执行代码
 - 3.4.2 翻牌后延迟代码的执行
 - 3.4.3 在JavaScript中随机化数组
 - 3.4.4 通过HTML5自定义数据属性保存内部自定义数据
 - 3.4.5 用jQuery访问自定义数据属性
 - 3.4.6 制作其他纸牌游戏
 - 3.5 在游戏中嵌入Web字体
 - 3.6 总结
- 第4章 用Canvas和绘图API构建《解题》游戏
 - 4.1 介绍HTML5 Canvas元素
 - 4.2 在Canvas中绘制圆形
 - 4.2.1 当Web浏览器不支持Canvas时的反馈信息
 - 4.2.2 用Canvas的arc函数绘制圆和图形

<<HTML5游戏开发实战>>

- 4.2.3 把角度转换为弧度
- 4.2.4 在Canvas中执行路径绘制的操作
- 4.2.5 在切换路径样式时先调用BeginPath
- 4.2.6 关闭路径
- 4.2.7 将画圆功能封装成函数
- 4.2.8 在JavaScript中生成随机数
- 4.2.9 保存圆的位置
- 4.3 在Canvas中画线
- 4.4 通过Canvas中的鼠标事件与绘制对象交互
 - 4.4.1 在Canvas元素中获取鼠标位置
 - 4.4.2 在Canvas中检测鼠标事件是否发生在圆上
 - 4.4.3 游戏主循环
 - 4.4.4 清除Canvas
- 4.5 在Canvas中检测线的交点
- 4.6 制作《解题》游戏
 - 4.6.1 定义关卡数据
 - 4.6.2 判断是否过关
 - 4.6.3 显示当前关卡和完成进度
- 4.7 总结
- 第5章 构建大师级Canvas游戏
 - 5.1 用渐变色来填充图形
 - 5.1.1 给渐变添加色标
 - 5.1.2 填充径向渐变色
 - 5.2 在Canvas中绘制文本
 - 5.3 在Canvas中绘制图像
 - 5.3.1 使用drawImage函数
 - 5.3.2 装点Canvas游戏
 - 5.4 在Canvas中播放精灵表动画
 - 5.5 创建多层Canvas游戏
 - 5.6 总结
- 第6章 给游戏添加声音效果
 - 6.1 给PLAY按钮添加声音效果
 - 6.1.1 定义audio元素
 - 6.1.2 播放声音
 - 6.1.3 暂停声音
 - 6.1.4 调整音量大小
 - 6.1.5 使用jQuery的hover事件
 - 6.1.6 创建Ogg格式的音频以支持Mozilla Firefox
 - 6.1.7 不同Web浏览器所支持的音频格式
 - 6.2 构建迷你钢琴音乐游戏
 - 6.2.1 在HTML5游戏中创建场景
 - 6.2.2 让音乐播放可视化
 - 6.2.3 为音乐游戏选择正确的歌曲
 - 6.2.4 存储和提取歌曲关卡数据
 - 6.2.5 获取游戏的流逝时间
 - 6.2.6 创建音乐点
 - 6.2.7 移动音乐点

<<HTML5游戏开发实战>>

6.3 将PLAY按钮与音乐游戏场景链接起来

6.4 构建键盘驱动的迷你钢琴音乐游戏

6.4.1 通过按键来击打三条音乐线

6.4.2 判断按键是否击中音乐点

6.4.3 通过给定索引将元素从数组中移除

6.5 给迷你钢琴游戏添加额外的功能

6.5.1 根据玩家的表现而调整音乐音量

6.5.2 从游戏中删除音乐点

6.5.3 保存最近5个音乐点的成功率计数

6.5.4 记录音符来得到关卡数据

6.6 音乐播放完后处理audio事件

6.7 总结

第7章 利用本地存储技术保存游戏数据

7.1 使用HTML5本地存储技术保存数据

7.1.1 创建游戏结束对话框

7.1.2 在浏览器中保存成绩

7.1.3 通过本地存储技术保存和加载数据

7.1.4 本地存储只保存字符串值

7.1.5 将本地存储对象看做关联数组

7.2 在本地存储中保存对象

7.2.1 在JavaScript中获取当前日期和时间

7.2.2 使用原生JSON将对象编码成字符串

7.2.3 从JSON字符串中加载所保存的对象

7.2.4 在控制台窗口中检测本地存储

7.3 用一种漂亮的彩带效果来告诉玩家破记录了

7.4 保存整个游戏的进度

7.4.1 保存游戏进度

7.4.2 从本地存储中删除记录

7.4.3 在JavaScript中复制数组

7.4.4 恢复游戏进度

7.5 总结

第8章 利用WebSocket构建多人游戏—《我画你猜》

8.1 初试WebSocket Web应用程序

8.2 安装WebSocket服务器

8.2.1 安装Node.js WebSocket服务器

8.2.2 创建广播连接数的WebSocket服务器

8.2.3 初始化WebSocket服务器

8.2.4 在服务器端监听连接事件

8.2.5 在服务器端获取已连接的客户端数

8.2.6 向所有已连接的浏览器广播消息

8.2.7 创建客户端来连接WebSocket服务器并获取总连接数

8.2.8 建立WebSocket连接

8.2.9 WebSocket客户端事件

8.3 使用WebSocket构建聊天室

8.3.1 向服务器发送消息

8.3.2 从客户端发送消息到服务器

8.3.3 在服务器端接收消息

<<HTML5游戏开发实战>>

- 8.4 通过在服务端广播接收到的消息来创建聊天室
- 8.5 使用Canvas和WebSocket制作共享绘图板
 - 8.5.1 构建本地绘图板
 - 8.5.2 广播绘图数据给所有已连接的浏览器
 - 8.5.3 定义用于在客户端与服务器之间进行通信的数据对象
 - 8.5.4 将画线数据打包成JSON以便于广播
 - 8.5.5 再现从其他客户端接收到的画线数据
- 8.6 构建多人游戏：《我画你猜》
 - 8.6.1 控制多人游戏的游戏流程
 - 8.6.2 在服务器端罗列出所有已连接客户端
 - 8.6.3 在服务器端发送消息给指定的连接
 - 8.6.4 改进现有游戏
- 8.7 用CSS装点《我画你猜》游戏
- 8.8 总结
- 第9章 用Box2D和Canvas构建物理类汽车游戏
 - 9.1 安装Box2d JavaScript库
 - 9.1.1 用b2World创建新的物理世界对象
 - 9.1.2 用b2AABB定义物理边界
 - 9.1.3 给物理世界设置重力
 - 9.1.4 设置 Box2D忽略休眠物体
 - 9.2 在物理世界中创建静态地面
 - 9.2.1 创建物体形状
 - 9.2.2 创建物理物体
 - 9.3 在Canvas上绘制物理世界
 - 9.4 在物理世界里创建动态的长方形
 - 9.5 推进物理世界的时间
 - 9.6 给游戏安装车轮
 - 9.7 创建物理汽车
 - 9.8 通过键盘给汽车施加动力
 - 9.8.1 向物体施加动力
 - 9.8.2 理解ApplyForce与ApplyImpulse之间的不同点
 - 9.8.3 给游戏环境添加坡道
 - 9.9 在Box2D世界检测碰撞
 - 9.10 重启游戏
 - 9.11 让游戏支持关卡
 - 9.12 为Box2D绘制图形轮廓
 - 9.12.1 使用形状和物体的userData属性
 - 9.12.2 根据物理物体的状态来绘制每一帧图像
 - 9.12.3 在Canvas中旋转和翻转图片
 - 9.13 给游戏添加装饰，让游戏更具趣味性
 - 9.13.1 用燃料的限制加速
 - 9.13.2 用CSS3进度条显示剩余燃料
 - 9.14 总结
 - 9.14.1 HTML5 游戏引擎
 - 9.14.2 游戏精灵以及贴图
 - 9.14.3 声音效果
- 附录 突击测验答案

<<HTML5游戏开发实战>>

章节摘录

这是因为当调用第2条fill命令时，Canvas中的路径列表还包含两个圆。因此，fill命令用绿色填充这两个圆，也就是重新填充原来红色的圆。

为了解决这个问题，需要确保在每次绘制一个新形状前都调用beginPath。

beginPath清空路径列表，所以下次调用fill和stroke命令时，它将只应用上次调用beginPath之后的所有路径。

牛刀小试 刚才我们讨论了一段代码，其中原打算绘制两个圆并分别用红色和绿色填充，但却绘制了两个绿色的圆。

怎样添加beginPath命令才能让代码按原来的打算来绘制圆呢？

4.2.6 关闭路径 closePath函数将会从最新路径的终点到路径的起点之间绘制一条直线，用于闭合路径。

如果我们只打算填充路径而不打算对其进行描边，closePath函数将不会影响结果。

对比同一个半圆的两种结果，其中一个调用closePath而另一个没有调用closePath函数。

.....

<<HTML5游戏开发实战>>

编辑推荐

《HTML5游戏开发实战》清晰而全面地展示了如何使用最新的HTML5 和CSS3 标准来构建纸牌、绘图、物理等各种常见类型的游戏。

<<HTML5游戏开发实战>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>