

HTML5 新技术的应用设计与实现技巧

姜照昶¹, 苏宇¹, 丁凯孟²

(1. 中国电子科技集团 第五十四研究所, 河北 石家庄 050081;

2. 中国科学院地理科学与资源研究所 资源与环境信息系统国家重点实验室, 北京 100101)

摘要: HTML5 是最新一代的 HTML 规范, 在支持 HTML5 的浏览器上可以不使用插件时实现许多实用且强大的功能。文中利用 HTML5 的 Canvas (画布)、WebSocket (实时通信)、Audio (音频) 技术及 (javascript) JQuery 库编程方法, 设计并实现一个基于 Web 的中英文打字游戏软件。该软件具有用户设置难易度、分数等级、音乐开关、中文/英文单词或文本选择、动态游戏场景选项、历史数据存储与成绩查询等功能, 同时利用 HTML5 的若干新技术和新特性如 Canvas 绘图 API、轻量型动画引擎、localStorage (本地存储) 结合 CSS 和 JQuery 库的技巧性设计, 实现了控制页面的锚定布局和动画特效。用 JavaScript 实现游戏算法, Access 创建数据库存储单词, 解决了一般打字游戏中的文本单词选择性差等问题。简洁有趣的字符飞行、消失等动画界面, 软件功能实用、界面友好、操作简便, 支持多种浏览器运行环境。

关键词: HTML5; 绘图画布; 本地存储; Ajax

中图分类号: TP399

文献标识码: A

文章编号: 1673-629X(2019)12-0167-06

doi: 10.3969/j.issn.1673-629X.2019.12.030

Design Techniques and Applied Implementation with HTML5

JIANG Zhao-chang¹, SU Yu¹, DING Kai-meng²

(1. The 54th Research Institute, China Electronic Technology Group Corporation, Shijiazhuang 050081, China;

2. State Key Laboratory of Resource & Environment Information System, Institute of Geographic Sciences and Natural Resources Research, CAS, Beijing 100101, China)

Abstract: HTML5 is the latest generation of the HTML specification in which browsers can implement many practical and powerful functions without plug-ins. In this project a Web-based typing game system is designed by using the HTML5 techniques, including Canvas (canvas), WebSocket (real-time communication), Audio (audio processing) and (javascript) JQuery library programming. It has some functions, such as difficulty setting, score level, the music switches, words or text selection, a variety of dynamic gaming scene design, historical data storage and query results, concise interesting characters flying, etc. In design process, we combine CSS with Canvas API, localStorage (local storage) and JQuery library to control page layout and special effects. Additionally, we use JavaScript to realize the game algorithm, Access to create a database to store words, and solve the poor selectivity of text words in general typing game. With simple and interesting animation interface of character flight and disappearing, the software has practical functions, friendly interface, easy operation and supports a variety of browser operating environment.

Key words: HTML5; canvas drawing; local storage; Ajax

0 引言

HTML 是用于描述网页文档结构的超文本标记语言, 万维网联盟 (W3C) 一直致力于发展 SGML (标准通用标记语言)、XML 以及一些新的标记语言如 SVG (可缩放矢量图形)、XForms 和 MathML 等^[1]。由于

HTML5 的标签表单、通信、离线应用、本地存储、设备访问和音视频多媒体等新特性的性能与表现, 使得 HTML5 发展成为构建 Web 应用的新一代标准和 HTML 规范, 在 Web 应用和移动互联领域得到了广泛的支持和发展^[2-3]。HTML5 拥有更丰富的标签, 支持

收稿日期: 2019-02-17

修回日期: 2019-06-20

网络出版时间: 2019-09-24

基金项目: 国家自然科学基金 (61375121, 41801303); 高层次引进人才科研项目 (jit-rcyj-201505, jit-b-201520); 江苏高校省级自然科学基金重大项目 (17KJA520001, 18KJA520003); 资源与环境信息系统国家重点实验室开放基金

作者简介: 姜照昶 (1986-), 男, 工程师, 研究方向为群智能计算与智慧软件工程等; 苏宇 (1993-), 男, 助理工程师, 通讯作者, 研究方向为智慧软件与无人机群智能控制规划等。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20190924.1537.056.html>

HTML5 的浏览器可以不使用插件来播放网页中的视频和音频,Canvas 标签为程序设计者提供了绘图 API,本地存储功能可使用户更快地打开应用程序等^[4]。HTML5 是 Web 技术中变革性的创新,打破了 Web 游戏由 Flash 控制的局面。研发人员使用 HTML5 开发了越来越多的交互性更强、效果更出众的 Web 应用和游戏^[5]。近年来,人们对 HTML5 的兴趣骤然爆发,涌现出大量支持 HTML5 的浏览器和设备^[6]。文献[7-8]分析了 HTML5 的 Web Socket 握手协议机制并在 Linux 平台环境下使用 Ruby 编程语言实现了一个简单的 Web Socket 服务器。文献[9]针对传统实时 Web 技术存在实时性差、吞吐量大的缺点,通过 Node.js 搭建 Web Socket 服务器,构建一种基于 HTML5 的实时 Web 数据监测系统,利用 Web Socket 通信机制自定义 Web Worker 接口,设计多服务计算迁移,有效提升 Docker Swarm 集群服务性能和智能终端 Web 应用性能。文献[10-11]采用 HTML5 Canvas 技术研究并实现了一个新型在线流程图协作绘制平台,将数据集中存储在云端,方便共享与协作。文献[12-14]采用 persistent publish/subscribe (PPS) 消息传递模型实现 HTML5 与硬件平台之间的通信,大大提高了 QNX 操作系统下车载嵌入式应用与硬件平台的数据交互的实时性。通过 HTML5 的 Canvas,使得用户不再需要安装插件或客户端软件即可实现时空联合目标轨迹动态的数据渲染和实时可视化处理^[15-17],设计具有几何建模物理特性和过程建模的,运行于浏览器上,具有强交互性、高仿真度的虚拟实验室。结合 CSS 和 JavaScript 设计的轻量型动画引擎,很简洁地实现了物理滤镜、锚定布局的艺术效果。同时 HTML5 新技术的安全问题得到了包括传统威胁延伸、恶意利用检测、跨平台安全性、新安全 Web 应用的跨站脚本动态检测和多浏览器漏洞挖掘和模糊测试^[18-19]。文中通过基于 Web 的中英文打字游戏主要功能的设计,分析 HTML5 新技术的应用设计与实现技巧。

1 主要功能实现技术及关键代码分析

1.1 CSS 与界面设计

层叠样式表(cascading style sheet,CSS)是用来显示 HTML 元素、用于控制应用程序的界面,并通过 CSS 实现了 Web 内容界面与其表现形式的分离处理,极大地提高了程序设计工作的效率。样式可以放在 3 个位置即样式表、<style>标签和 style 属性。样式表(Stylesheet)是一个独立文件,可以使用<link>元素和 CSS 的@import 语句将其附加到 HTML 文档。<style>是所有 HTML 元素都支持的元素,可以嵌入到 HTML 文档中。通常样式保存在样式表中,能够减少 HTML

文档中的非内容元素,并且将所有样式保存在文件中,方便管理。

一般用样式表的位置来控制其范围,如果一个样式表适用于整个网站,那么应该将它保存在网站的根目录。如果一个样式表只适用于某个文档,那么应该将其保存在该文档相同的目录中。另一种网站文件组织方式是将所有样式表保存在一个目录中。在 HTML 文档的<head>部分加入一个<link>元素,就可以将样式表链接到 HTML 文档,而<link>元素的 href 属性可以指定样式表的 URL。最新的 CSS3 增加了许多特性,利用新增的特性可以创造出许多绚丽的效果。打字程序所涉及的主要问题是字符串的处理,所以界面比较单调,点击相应的按钮可进入相应的游戏界面。界面上有一固定的按钮用来控制音乐的开关,其样式在 word.css 文件中,主界面效果如图 1 所示。

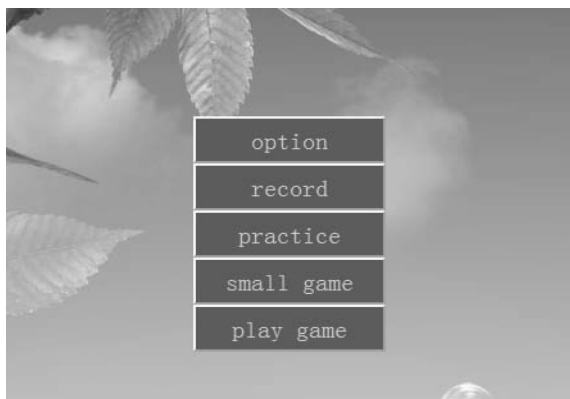


图 1 游戏主菜单界面

1.2 Ajax 与随机单词生成

所谓 Ajax 就是基于 XML 的异步 JavaScript,用于概括异步加载页面内容的技术。以前,Web 应用都要涉及大量的页面刷新:用户点击了某个链接,请求发送回服务器,然后服务器根据用户的操作再返回新页面。即使用户看到的只是页面中的一小部分有变化,也要刷新和重新加载整个页面,包括公司标志、导航、头部区域、脚部区域等。使用 Ajax 就可以做到只更新页面中的一小部分,其他内容都不用重新加载。用户仍然像往常那样点击链接,但这一次,已经加载的页面中只有一小部分区域会更新,而不必再次加载整个页面。

Ajax 的主要优势就是对页面的请求以异步方式发送到服务器。而服务器不会用整个页面来响应请求,在后台处理请求,与此同时用户还能继续浏览页面并与页面交互。脚本则可以按需加载和创建页面内容,而不会打断用户的浏览体验。Ajax 有它自己的适用范围。它依赖于 JavaScript,所以可能会有浏览器不支持它,而搜索引擎的蜘蛛程序也不会抓取到有关内容。

Ajax 的核心就是 XMLHttpRequest 对象。这个对

象充当着浏览器中的脚本(客户端)与服务器之间的中间人的角色。以往的请求都由浏览器发出,而 JavaScript 通过这个对象可以自己发送请求,同时也自己处理响应。该游戏软件设计中所使用的单词是从数据库中获取的。由于纯 JavaScript 连接数据库只能在 IE 下运行,且安全性低。因此该程序使用 Ajax 技术与服务器进行交互,使用 GET 方法传递给服务器一个随机 ID 号,然后获取该 ID 所对应的单词。下面的函数用于创建一个 XMLHttpRequest 对象以便与服务器进行交互,其具体代码如下:

```
function createXMLHttpRequest()  
{  
    if( window. XMLHttpRequest )  
    {  
        //识别浏览器  
        xmlhttp=new XMLHttpRequest();  
    }  
    else  
    {  
        // code for IE6, IE5  
        xmlhttp=new ctiveXObject( " Microsoft. XMLHttpRequest" );  
    }  
    //传递给数据库一个随机 ID 号,以便获取单词  
    function startRequest()  
    {  
        var i=parseInt(Math. random() * 10000)+1;  
        createXMLHttpRequest();  
        xmlhttp. onreadystatechange=processData;  
        xmlhttp. open ( " GET" , " wordlist. asp? name = " + i + " " ,  
true );  
        xmlhttp. send ( );  
    }  
    //接收从服务器获取的单词  
    function processData()  
    {  
        if( xmlhttp. readyState == 4 && xmlhttp. status == 200 )  
        {  
            backData = xmlhttp. responseText;  
            temparr = backData. split( " ?" ) [ 1 ] ;  
            word = temparr. split( " =" ) [ 0 ] ;  
            mean = temparr. split( " =" ) [ 1 ] ;  
            randword( word, mean );  
        }  
    }  
}
```

1.3 JQuery 库与字符判断

JQuery 是一种兼容 CSS3 的轻量级 JavaScript 库,与多种浏览器包括 IE 8.0+、FF 1.5+、Safari 2.0+、Opera 9.0+等具有很好的兼容性。JQuery 库使用户能

更方便地处理 Web 页文档、事件以及各种特效。在 JavaScript 出现之后,Web 网页的内容不再局限于枯燥的文本,它们的交互性得到了显著改善。JavaScript 是一种解释性的脚本语言,通常只能通过 Web 浏览器去完成一些操作而不能像普通意义的程序那样独立运行,因为需要有 Web 浏览器进行解释执行。JavaScript 的强大之处在于它可以操作 DOM,从而产生多种 JavaScript 特效。

游戏需要解决的问题是字符串的处理,而 JavaScript 的 String 对象提供了许多方法和属性用于处理字符串,打字游戏中对输入字符的判断和识别是关键技术,该函数用于判断用户输入的空缺字符是否正确。

```
function judge()  
{  
    var flag=0;  
    for( var i=0;i!=charNumber;i++)  
        if( document. getElementById( " " + i + " " ). value == tempChar  
[ i ] )  
        {  
            flag = 1;  
            else flag=0;  
            if( flag )  
            {  
                remainWords=remainWords-1;  
                if( remainWords == -1 )  
                {  
                    clearTimeout( T );overMenu();return;  
                }  
                rw. innerHTML=" remain words:" +remainWords;  
                corWords=corWords+1;  
                cw. innerHTML=" correct words:" +corWords;  
                startRequest();  
            }  
        }  
}
```

在主界面点击 smallgame 按钮即可进入该模式,游戏开始后,屏幕上方会随机掉落一些字母或数字,用户敲击键盘上相应的键,会消除字母。当这些字母或数字块掉落到屏幕底部时,用户还未将其消除,则生命值减一,当生命值降为零,游戏结束,程序会弹出对话框显示用户得到的分数。点击 BACK 会返回程序主界面。此模式中还有一个 CLEAR 按钮,点击此按钮,会清除屏幕上现有的所有字母数字块。

对于按键事件 onkeydown,由于 FireFox 和 IE 的处理函数不同,因此需分别处理。在 FireFox 中需要一个参数,响应按键字符的是 which 属性在 IE 及其他浏览器中不需要此参数,响应案件字符的是 event. keyCode。此模式需要处理的问题是,用户按键的判断,字母数字块的正确消除。解决的方法是每当创建

一个字母数字块时,将其推入一个数组中,当此块降到屏幕底部时,将其从数组中删除,而当用户敲击一个正确的字符时,也将其从数组中删除。在 HTML 中<button onclick=" miniStart ()" >small game</button>,此函数用于创建字母或数字块,并显示动画效果,如图 2 和图 3 所示。

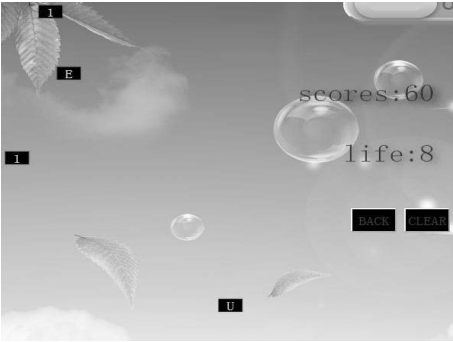


图 2 游戏 Play 界面



图 3 练习模式中的系统界面

其关键技术 JS 代码如下:

```
function miniCreate()  
{  
    i=i+1;  
    var index=alpha[ parseInt( Math. random( ) * alpha. length) ]  
    var xpos=parseInt( Math. random( ) * 800)+100;  
    var miniDiv=document. createElement( " div" );  
    miniDiv. setAttribute( " id", i );  
    document. body. appendChild( miniDiv );  
    miniDiv. style. position=" absolute";  
    miniDiv. style. left=xpos+" px";  
    miniDiv. style. top=" 40px";  
    miniDiv. style. width=" 30px";  
    miniDiv. style. height=" 20px";  
    miniDiv. style. border=" solid 1px red";  
    miniDiv. innerHTML=index;  
    miniDiv. style. fontSize=" 20px";  
    miniDiv. style. textAlign=" center";  
    miniDiv. style. background=" black";  
    miniDiv. style. color=" white";  
    data. push( miniDiv );  
    var index=" #" +i;  
    $( index ). animate( { top: 600 },9000,function() {\[7]
```

```
}  
$( index ). remove( );  
life-= 1;  
lifes. innerHTML=" life:" +life;  
data. splice( index,1) } );  
if( life ==0)  
{  
    miniOver( );  
    alert( " Game over! Your scores:" +scores );  
    return;  
}  
MT=setTimeout( miniCreate, 1500 );  
}  
//判断用户按下的键与屏幕上现有的字母或数字快哪个相符  
function miniJudge()  
{  
    if( ! window. event )  
    {  
        //for FireFox browser  
        document. onkeydown= function( event ) {  
            var k=event. which;  
            for( var j=0;j! =data. length;j++)  
            if( String. fromCharCode( k) = =data[ j ]. innerHTML )  
            {  
                document. body. removeChild( data[ j ] );  
                $( data[ j ] ). stop( );  
                data. splice( j,1 );  
                scores+= 10; score. innerHTML=" scores:" +scores;  
            }  
        }  
    }  
    else  
    {  
        //for other browser  
        document. onkeydown= function( event )  
        {  
            var k=window. event. keyCode;  
            for( var j=0;j! =data. length;j++)  
            if( String. fromCharCode( k) = =data[ j ]. innerHTML )  
            {  
                document. body. removeChild( data[ j ] );  
                $( data[ j ] ). stop( );  
                data. splice( j,1 );  
                scores+= 10;  
                score. innerHTML=" scores:" +scores;  
            }  
        }  
    }  
}
```


1.4 localStorage 与模式设置

localStorage(本地存储):对于统计分数创建一个数据库来存储,但只有一个变量,觉得用数据库有点麻烦,于是就改用html5的一个新特性:localStorage(本地存储)功能,它是用来代替以往的 cookie。但是这一特性有一个 BUG,可以占满用户的硬盘,是国外某位网友利用一个 API 发现的。要想正确使用 localStorage 这一特性,必须确保浏览器要设置不会清除缓存,因为 localStorage 本质上是浏览器储存数据和 cookie 是一样的。

游戏模式:当用户设置好单词数和时间后点击 play game 按钮时,程序进入此模式。此时屏幕会随机显示一个从数据库中取到的单词和释义。并将该单词的部分字母去掉并用一个下划线代替,用户可在空缺的地方填入相应的字母,如果全部填正确,会自动生成下一个单词。当时间结束或单词全部拼完,会显示用户的成绩,点击 main menu 按钮,程序返回主界面。游戏中当用户遇到不会拼写的单词时可随时点击 NEXT 按钮,跳过当前单词并生成另一个单词,或者点击 BACK 按钮结束游戏返回主界面。

选项设置:点击 option 按钮进入设置界面,该选项只对游戏模式有效,用户可在此设置单词数和拼完这些单词可用的时间。必须同时输入正确的时间和单词数,只能是数字,如果输入错误或者有一项没输,则会提示错误,效果图如图 4 所示。

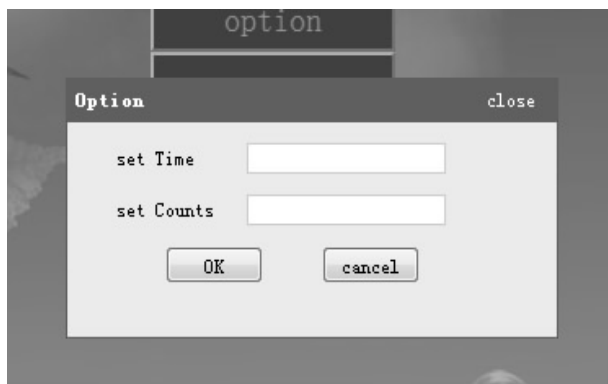


图 4 选项设置界面

2 游戏中的数据库设计

2.1 单词表设计

游戏中设计一个单词数据库,设计一个表,表名为 cetsix,有三个字段意义,如表 1 所示。

表 1 单词表 cetsix 字段类型及其意义

字段名	数据类型	意义
ID	Varchar(50) NOT NULL	单词编号,主键
words	Varchar(50) NOT NULL	单词
meanings	Varchar(100) NOT NULL	单词释义

2.2 数据库连接

HTML5 规范中进行 Web SQL 数据库 API 操作,它通过一套 API 来操纵客户端的数据库。Safari、Chrome、Firefox、Opera 等主流浏览器都已经支持 Web SQL Database^[9],需要经历创建打开数据库,创建表,添加数据,更新数据,删除数据,删除表等数据操作。其核心方法是:openDatabase、transaction、executeSql 分别用于创建数据库对象、控制事务提交或回滚及执行真实的 SQL 查询。游戏中数据库操作的关键技术 JS 实现代码如下:

```
var dataBase = openDatabase( " student", " 1. 0", " 学生表",
1024 * 1024,function() {});
if(! dataBase)
{
//打开连接并创建数据库
alert(" 数据库创建失败!");
}
else
{
alert(" 数据库创建成功!");
}
<% Response. Addheader" Content-Type", " text/html;
charset=gb2312" %>
<% dim i i=request.querystring( " name" )
//获取 JS 文件传来的参数
set conn=Server. CreateObject( " ADODB. Connection" )
conn. Provider=" Microsoft. Jet. OLEDB. 4. 0"
conn. Open " englishword. mdb"
set rs=Server. CreateObject( " ADODB. recordset" )
rs. Open " Select * from cetsix where ID=" & i & " ", conn
response. write( "<br>" & "?" & rs( " words" ) & " =" & rs( "
meanings" )) %>
```

2.3 Ajax 与服务器交互

利用 Ajax 与服务器交互需要先创建一个 XMLHttpRequest 的对象 (IE8 和 IE9 需要创建 ActiveXObject 对象),此对象可以在后台和后台中的数据库进行数据交换。也就是说不需要加载所有页面内容,只更新一部分的网页内容,即进行局部刷新。构建一个 XMLHttpRequest 的对象:variable = new XMLHttpRequest(),向服务器发送请求:如需将请求发送到服务器,使用 XMLHttpRequest 对象的 open() 和 send() 方法:open(method,url,async);send(string),open 方法中的 method 参数代表请求的类型,可以是 GET 或 POST,url 代表文件在服务器上的位置,async 代表是否是异步请求;send 方法用于将请求发送到服务器,参数 string 仅用于 POST 方式。

服务器响应:如果需要获取来自服务器的响应请

求,需要使用 XMLHttpRequest 对象的 responseText 属性或者是 responseXML 属性;responseText 这一属性用于获取字符串类型的响应;responseXML 这一属性用于获取 XML 类型的响应。该程序使用 responseText 属性以获取字符串形式的响应 onreadystatechange 事件,当请求被发送到服务器时,基于响应的一些任务需要被执行。每次当 readyState 属性被改变时,就会触发一个事件,方法名为 onreadystatechange。XMLHttpRequest 的一些信息状态是存储在 readyState 属性中的。

3 结束语

设计并实现了一个基于 Web 的中英文打字游戏软件,具有用户设置难易度、分数等级、音乐开关、中文/英文单词或文本选择、动态游戏场景选项、历史数据存储与成绩查询等功能,同时对 Canvas 绘图 API、WebSocket(实时通信)、localStorage(本地存储)结合 CSS 和 JQuery 库的技巧性设计,实现了控制页面的布局和特效,用 JavaScript 实现游戏算法和单词库,解决了一般打字游戏中的文本单词选择性差等问题。游戏服务端运行硬件配置要求低,简洁有趣的字符飞行、消失等动画界面,软件功能实用、界面友好、操作简便,支持多种浏览器运行环境,经测试可在 IE、FireFox、Chrome、Safari、Opera 等多个支持 HTML5 的浏览器上运行。

参考文献:

- [1] SHANKAR A R. Pro HTML5 games; learn to build your own games using HTML5 and JavaScript [M]. [s. l.]: Springer, 2017.
- [2] GUTIÉRREZ R T. Understanding the role of digital commons in the web; the making of HTML5 [J]. Telematics & Informatics, 2018, 35(5): 1438–1449.
- [3] KAPITSAKI G M, CHARALAMBOUS T. Adapting HTML5 Web applications to user privacy preferences [J]. World Wide Web, 2019, 22(5): 2041–2046.
- [4] 朱文. 基于 HTML5 Canvas 技术的在线图像处理方法的研究 [D]. 广州: 华南理工大学, 2017.
- [5] 周杨, 李燕, 李范鸣. 基于新型软件架构的 NFC 管理系统的设计实现 [J]. 计算机技术与发展, 2018, 28(2): 1–4.
- [6] 李营那, 吴松洋, 张涛, 等. 基于动态标签技术的信息发布系统设计与实现 [J]. 计算机应用与软件, 2017, 34(1): 21–27.
- [7] 郝立坤, 何小刚. 基于 HTML5 的虚拟实验建模及实现 [J]. 现代电子技术, 2018, 41(24): 66–70.
- [8] 王媛. 基于 HTML5 技术的时空联合目标轨迹动态可视化技术 [J]. 科学技术与工程, 2018, 18(29): 98–103.
- [9] 罗帅, 罗斌, 汪明云. jQuery 炫酷应用实例集锦 [M]. 北京: 清华大学出版社, 2018.
- [10] YU W, XU Y. Research on the application of HTML5 in the development of mobile internet [J]. Journal of Intelligent & Fuzzy Systems, 2018, 34(2): 945–954.
- [11] 刘于沛. 基于 HTML5 进行响应式 Web 应用的技巧 [J]. 中国新通信, 2017, 19(2): 95.
- [12] MCLAUGHLIN B. Head rush AJAX [M]. Taiwan: O'Reilly Media, 2017.
- [13] 王昭, 邓浩江, 胡琳琳, 等. 一种多服务端 HTML5 Web Worker 迁移系统设计与实现 [J]. 计算机应用与软件, 2018, 35(9): 27–31.
- [14] 钟玲玲, 黄鹏, 黄靓, 等. 基于 HTML5 的定制糕坊系统设计 [J]. 佳木斯大学学报: 自然科学版, 2019, 37(1): 43–45.
- [15] XIE Ning, REN Mengyuan, YANG Wu, et al. WebPainter: collaborative stroke-based rendering through HTML5 and WebGL [C]//International conference on technologies for e-learning and digital entertainment. Bournemouth, UK: Springer, 2017: 210–217.
- [16] KAUR G, PANDE B, BHARDWAJ A, et al. Efficient yet robust elimination of XSS attack vectors from HTML5 web applications hosted on OSN-based cloud platforms [J]. Procedia Computer Science, 2018, 125: 669–675.
- [17] 张玉清, 贾岩, 雷柯楠, 等. HTML5 新特性安全研究综述 [J]. 计算机研究与发展, 2016, 53(10): 2162–2171.
- [18] 霍玮, 戴戈, 史记, 等. 基于模式生成的浏览器模糊测试技术 [J]. 软件学报, 2018, 29(5): 1275–1287.
- [19] 马富天, 钱雪忠, 宋威. 一种自动化的跨站脚本漏洞发现模型 [J]. 计算机工程, 2018, 44(8): 167–173.