

安徽科技学院

本科生毕业论文（设计）



题 目: 基于 HTML5 的 Web 交互游戏的设计与实现

姓 名: 董园原

学 院: 数理与信息工程学院

专 业: 计算机科学与技术

班 级: 2011 级 3 班

学 号: 1881110306

指导教师: 赵 靖 职称: 讲 师

2014 年 12 月 21 日

安徽科技学院教务处制

目 录

摘要	1
关键词	1
引言	1
1 基于 HTML5 的 web 交互游戏概述	1
1.1 研究目的和研究意义	1
1.2 研究现状和发展趋势	1
2 开发语言简介	2
2.1 CSS3 简介	2
2.2 jQuery 简介	2
2.3 HTML5 简介	2
2.4 HTML5 新 API	2
2.5 HTML5 WebSocket 简介	3
2.6 后台语言 Node.js	3
3 系统分析与设计	3
3.1 需求分析	3
3.1.1 需求说明	3
3.1.2 用例分析图	3
3.2 整体功能设计	4
3.2.1 程序功能分析	4
3.2.2 程序流程设计	4
4 详细设计与实现	6
4.1 数据库设计	6
4.1.2 数据库表	7
4.2 客户端设计	7
4.2.1 用户注册	7
4.2.2 用户登录	8
4.2.3 用户聊天	9
4.2.4 音乐设置	10
4.3 服务器端设计	10
4.3.1 启动服务器和 socket	11
4.3.2 用户注册	11
4.3.3 用户登录处理	12
4.3.4 进入房间	12
4.3.5 先手处理	13
4.3.6 用户游戏	13
4.3.7 退出房间	14
4.4 界面设计	15
5 总结	16
致谢	16
参考文献	17
英文摘要	18

基于 HTML5 的 web 交互游戏的设计与实现

计算机科学与技术专业学生 董园原

指导老师 赵靖

摘要：HTML5从流行到规范制定这个过程中，人们用它来开发游戏渐渐增多，甚至有人预言它将替代flash。在游戏领域HTML5的优势在于能够实现跨平台游戏编码移植^[3]；另外，HTML5支持的WebSocket API，这也使得HTML5成为游戏开发，特别是移动游戏平台的重要趋势。本文将介绍一款网页版的五子棋游戏，主要功能包括在线聊天，听音乐和玩游戏，用户无需下载APP，打开浏览器就可以游戏。本文采用Node.js作为开发环境，采用mongodb建立简单的数据库，网页美化利用HTML5+CSS3+jQuery，实时信息交互通过WebSocket实现，操作简单，界面简洁美观。

关键词：HTML5;Node.js;WebSocket;实时性;交互游戏;

HTML5语言开发游戏，在当下虽然非常有争议，但人们还是对它充满期待，HTML5对WebSocket API的支持也使得HTML5有更广阔的前景。综合来看，对于要求信息实时交互的网站，大部分都是采用HTTP的web实时机制，这种机制往往很复杂，所以简单的实时机制就显得尤为重要。新兴的WebSocket是一个基于TCP连接的全双工通信方式，服务端和客户端可以相互推送数据。一旦连接成功后，就可以在全双工的模式下在客户端和服务端之间来回传送WebSocket消息^[4]。它是HTML5的一个特性，将改变开发应用程序的方法，对于新特性，我们需要大胆去尝试它的好与坏。本文就是利用这个特性，结合异步式事件驱动的后台语言Node.js开发的五子棋游戏，尝试WebSocket与传统实时机制的优劣。

1 基于 HTML5 的 web 交互游戏概述

1.1 研究目的和研究意义

目前的互联网已经不仅仅是传统意义上的互联网，所有的传统应用和模式正在向着更加移动和便捷方向发展，人们可能通过手机、平板电脑等各类移动设备来获取信息和使用更加丰富的应用^[1]，未来必将也会改变人们生活的方方面面。

随着HTML5技术的日益完善，HTML5在游戏开发领域越来越受欢迎^[7]。HTML5游戏平台相比于传统游戏，具有更广阔的推广模式，它不需要下载APP，是直接基于网页和浏览器运行^[2]。Web是唯一一个适合各种应用广泛使用的运行环境，只要浏览器支持，HTML5游戏就能玩，这对于玩家是非常方便的，而我们就是需要追求这种便捷。

1.2 研究现状和发展趋势

HTML发展的飞跃是20世纪90年代，HTML版本经历了2.0、3.2到4.0，它的控制权在发展中慢慢由W3C（万维网联盟）掌控。2004年，Web超文本应用技术工作组（Web Hypertext Application Technology Working Group）成立，这是为了将Web平台提升至一个新的高度。这一小组人创立了HTML5规范，并且被Web超文本应用技术工作组认为是HTML中最薄弱的环节同时也在努力被开发。Web的第二时代--Web 2.0诞生了。旧的静态网站逐渐淘汰，具有更多特性的动态网站

和社交网站兴起^[3]，这些网站中加入了很多惊人的新功能。

万维网联盟重新介入HTML进行修改已经是2006年了，2008年，W3C正式发布了HTML5的工作草案。第二年，XHTML2工作组便停止工作。2010年，各大浏览器厂家开始升级自己旗下的产品，就是为了支持HTML5新功能。虽然此时HTML5规范还没有定稿，但是很多实际问题HTML5都能解决，因此，从这年开始，HTML5如雨后春笋般快速崛起^[3]。由于这些按耐不住的浏览器厂商的实验性反馈，HTML5才能在发展中不断完善自己的规范，2014年10月29日，W3C正式宣布HTML5规范完成。在此之前，HTML5的使用已经达到惊人的数量，据统计2013年全球能支持HTML的手机浏览器将有10亿的数量^[10]。在未来，HTML5不仅将用于企业模块，对于“云+终端”的架构，它也将改变自己的兼容方案^[6]。对于未来互联网发展，我们拭目以待。

2 开发语言简介

本案例前台语言采用了 HTML5+CSS3+jQuery，后台采用异步式事件驱动语言 Node.js，以下将进行逐一介绍。

2.1 CSS3 简介

CSS即层叠样式表(Cascading StyleSheet)，是用来在网页制作时有效地控制页面的效果，即装饰网页，使其变得更加漂亮。CSS3是CSS技术的升级版本，它的特点是与以往规范作为一个模块相反，它被划分成多个模块^[8]。这样，新的模块也在发展中不断加入进来。在兼容上CSS3是完全向后兼容，所以设计无需修改。对于CSS3模块分为如下几个：（1）选择器；（2）盒模型；（3）背景和边框；（4）文字特效；（5）2D/3D转换；（6）动画；（7）多列布局；（8）用户界面。相信更多新的模块也会很快加入。

2.2 jQuery 简介

jQuery 是一个兼容多浏览器的 JavaScript 库。原生 js 虽然可以写出很多很好的特效，但是代码冗长，写起来不方便，而 jQuery 的核心理念正是 write less,do more(写得更少,做得更多)，因此迅速流行起来。jQuery 学习起来非常简单，它的文档很全，各种应用非常全面而且详细，按照文档要求基本就可以掌握。它兼容 CSS3，还兼容各种浏览器，同时还有许多成熟的插件可供选择。jQuery 不需要和 HTML 代码放在一个文件中，分离后只需在 HTML 文件中引用即可^[9]。这种轻量级的 js 库可以方便地编写出高效代码，能让静态或动态页面更加吸引人。

2.3 HTML5 简介

HTML(Hyper Text Mark-up Language) 即超文本标记语言或超文本连接标示语言，不同的浏览器之间的信息交流就是通过 HTML 语言连接的。HTML5 是最新规范的 HTML 语言，于 2014 年 10 月 29 日为 W3C 宣布最终规范。

2.4 HTML5 新 API

除了原先的 DOM 接口，HTML5 增加了更多 API,如：

- 实时二维动画 Canvas API。
- 定时媒体播放 HTML5 音频与视频 API。
- 通信网络 Communication API。
- 地理 Geolocation API。
- 本地数据库 WebSQL Database，一个本地的 SQL 数据库^[6]。
- WebSocket，一个基于浏览器的 socket，支持实时性的数据传输^[4]。

- 其他：离线存储数据库（离线网络应用程序）、编辑、拖放等。

2.5 HTML5 WebSocket 简介

HTML5 规范引入了许多新功能，WebSocket 就是其新引入的一种通信协议，由通信协议和编程 API 组成。它主要是为了解决用于解决后台服务器与浏览器之间双向通讯的问题。WebSocket 与其他实时机制不同在于是一个基于 TCP 连接的全双工通信方式，服务端和客户端可以相互推送数据^[5]。也就是说，一旦连接成功后，前台和后台可以随时相互通信，保证前后台状态统一。在传统 HTTP 协议中，这是“无法做到”的。

2.6 后台语言 Node.js

JavaScript 作为前端语言，按理说是无法作为底层语言编写服务器的，但是 Node.js 改变了这一点。它提供给 JavaScript 语言运行在服务器端的开发平台，即是一种 JavaScript 的解析器。Node.js 的目标是帮助程序员构建高度可伸缩的应用程序，编写能够处理数万条同时连接到一个（只有一个）物理机的连接代码。与我们接触比较多的 PHP、java 不同，Node.js 的架构设计是异步式 I/O 与事件驱动。需要用到高并发的方案解决问题时，一般是采用传统的架构即多线程模型。但是为每个业务逻辑提供一个系统线程，会耗费很多系统开销。Node.js 使用的单线程模型，整个过程采用事件方式，它会将事件放在一个队列中，有条不紊地执行每个事件。

3 系统分析与设计

3.1 需求分析

3.1.1 需求说明

进入网站后，用户可以注册和登录，对于登录和注册需要有一定的限制。登录之后可以选择房间进入游戏（没有登录则无法进入游戏房间）。进入房间后，房间显示用户的昵称和头像。当房间只有一位用户时，用户是无法游戏的，但是可以听音乐。当房间有两个人时，网站首页对外显示该房间“正在战斗”，否则显示“可以进入”。游戏开始时，先手是第一个进入房间的人。先手点击“再来一盘”，方可进行游戏。游戏前，后手可申请先手，对方同意后，才能互换；游戏中，一方可以“认输”来结束游戏；游戏后，先手互换。在此过程中，双方可以相互聊天，输入的聊天内容不能为空。一旦用户进入房间，便可以点击“退出房间”。如果有一个用户退出房间，则另一方会收到对方退出的实时消息，并且清除对方的信息。

3.1.2 用例分析图

本文的用例分析图如下图 1 所示。

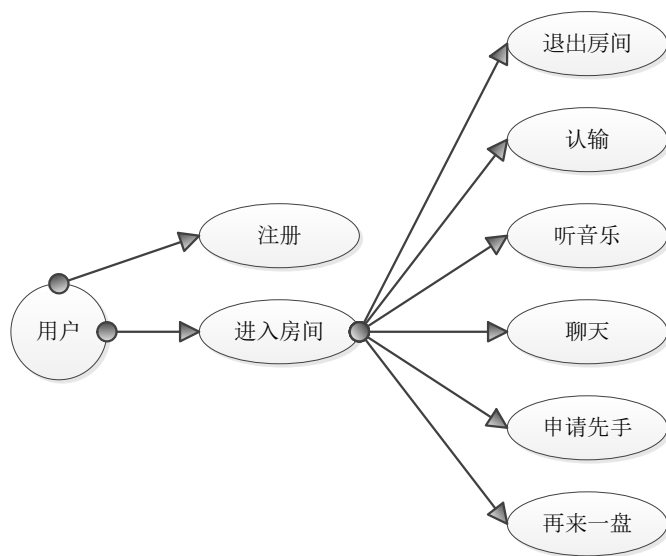


图 1 用例分析图

3.2 整体功能设计

3.2.1 程序功能分析

游戏房间有两人时，甲方点击棋盘的任一有效位置后，客户端需要向服务器端发送棋盘点击位置，服务器广播给所有用户客户端。乙方客户端通过筛选是否发送给自己的信息，若是就在发送过来的位置上放置棋子。筛选是为了避免别的房间用户发送信息到自己房间来，造成信息干扰。程序功能分析图如下图 2 所示。

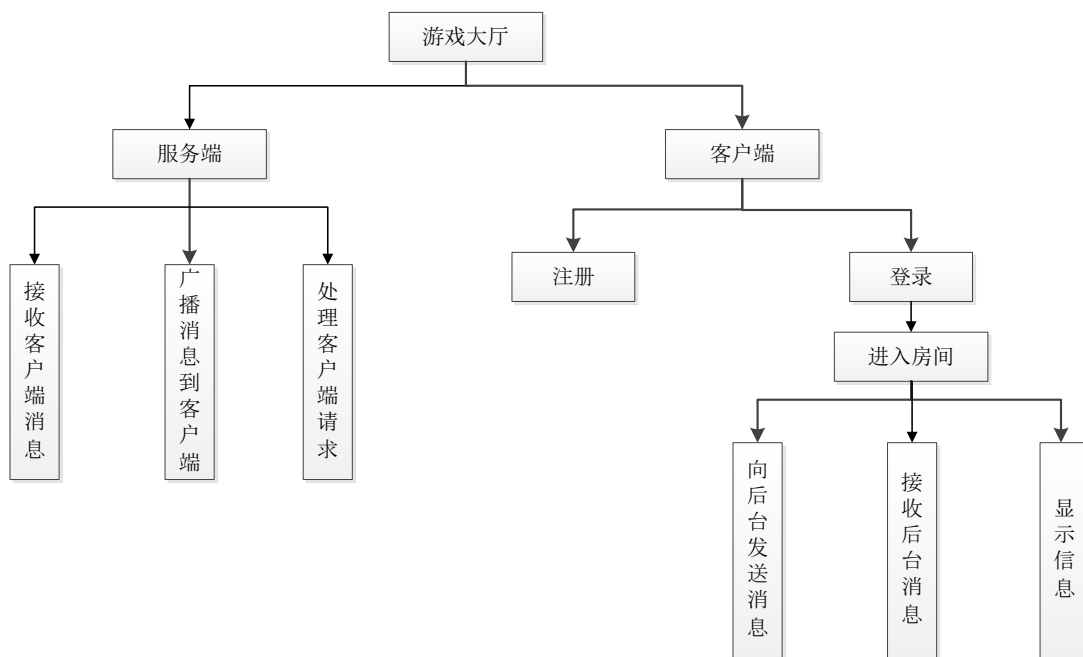


图 2 程序功能分析图

3.2.2 程序流程设计

(1) 用户注册流程。用户点击注册，则跳转至注册页面。对于注册的信息，在客户端需要先进行约束，才能注册。一旦注册成功，则在后台进行验证，通过才能存入数据库。流程图如图 3 所示。

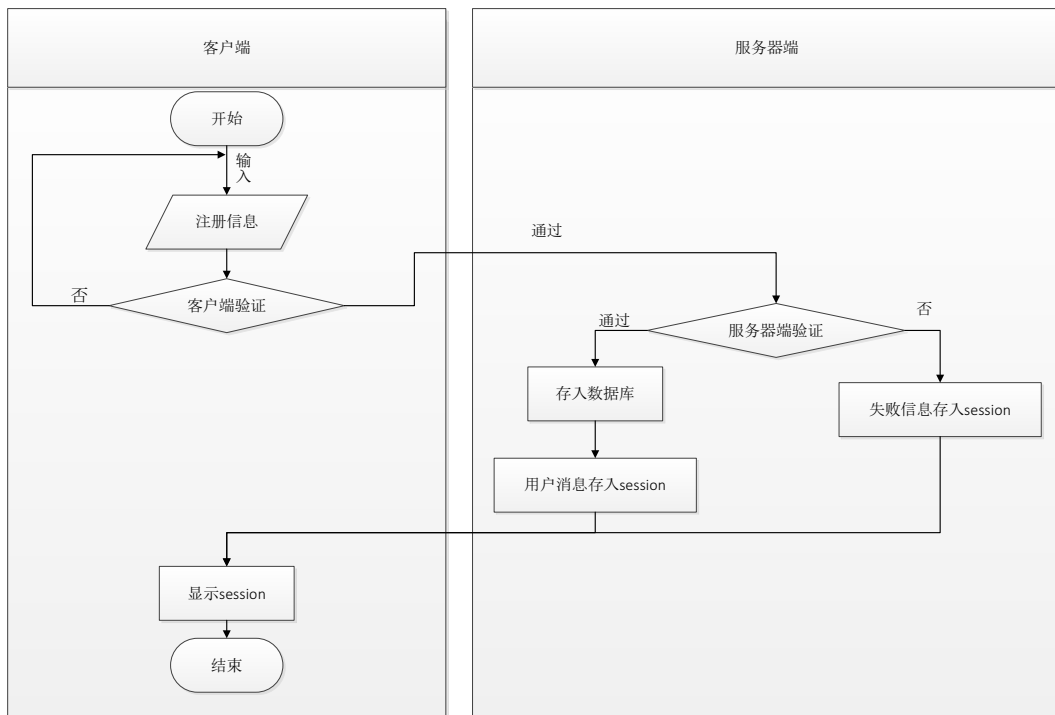


图 3 用户注册流程图

(2) 游戏流程。先手发送自己的棋盘位置到服务器端，在客户端需将已发送的位置标记为已有棋子位置，这样就不会在同一个位置点击而连续发送到后台。服务器会判断从这个位置的横向、竖向、左下和左上四个方向进行判断，是否有五子连续，如果是，说明游戏结束，否则，游戏继续，封装消息发送给前台即可。游戏流程图如下图 4 所示。

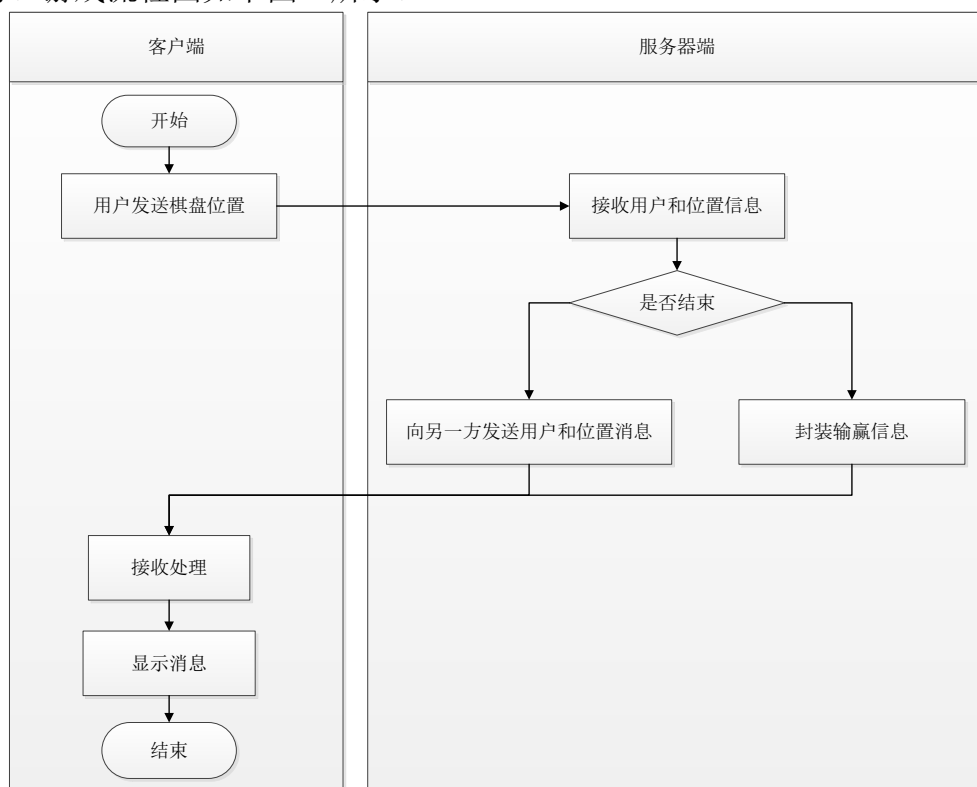


图 4 游戏流程图

(3) 聊天流程。用户在客户端的信息输入框中输入信息，客户端先判断是否有信息输入，有信息就会发送给服务器端，服务器将此消息广播给客户端，客户端根据接收的房间号进行筛选，是自己房间号的信息，就会显示在聊天框中。流程图如下图 5 所示。

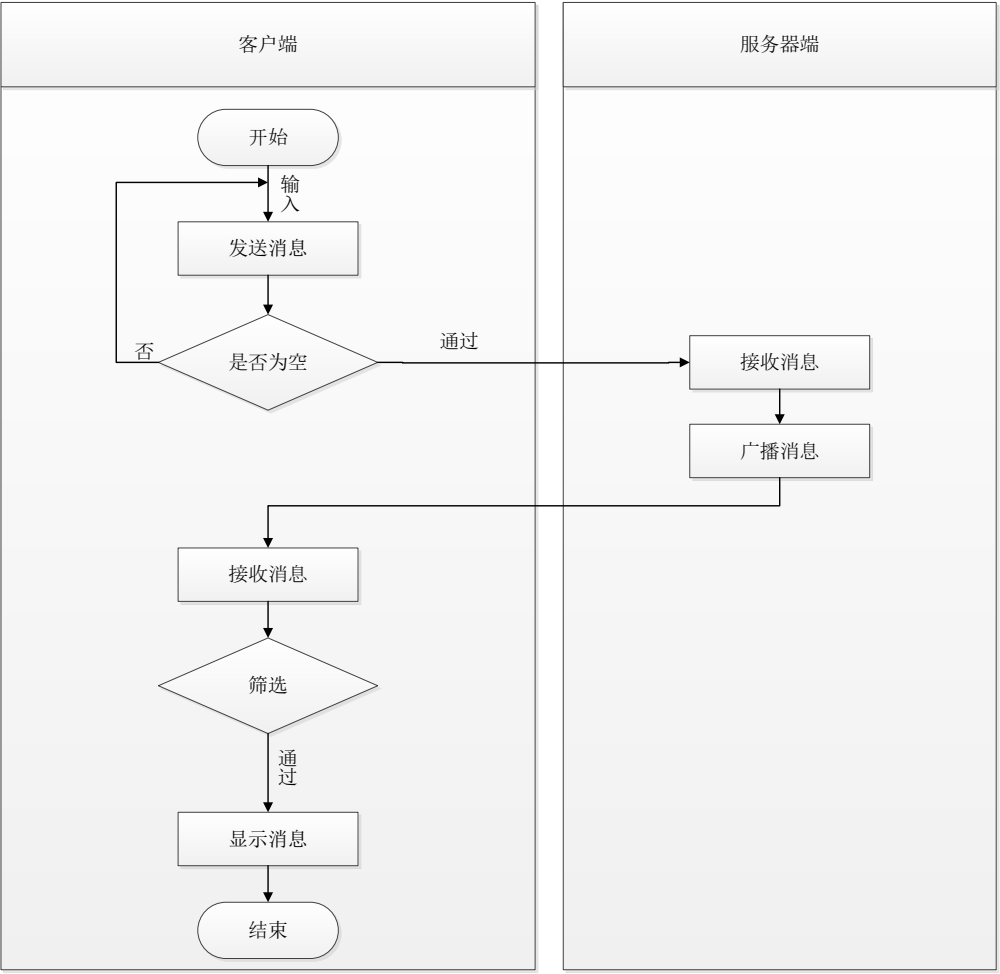


图 5 聊天流程图

4 详细设计与实现

4.1 数据库设计

本文采用基于分布式文件存储的 MongoDB 数据库设计了一个简单的数据库，存储了用户的信息。

4.1.1 E-R 图

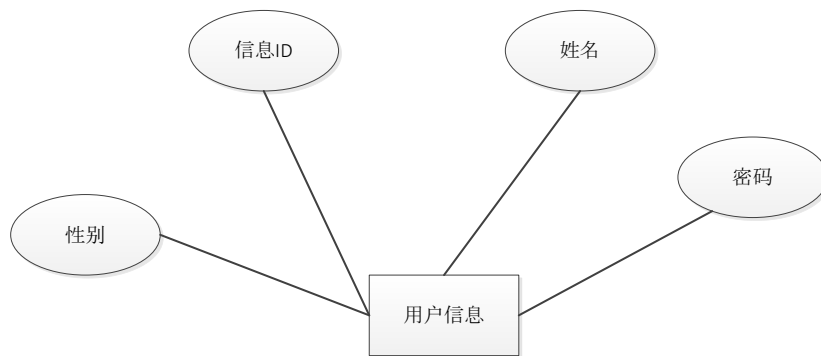


图 6 E-R 图

4.1.2 数据库表

根据需求，建立了 `users` 表，存放用户注册信息。在用户玩游戏时，需要一个 `id` 表示用户身份，以不至于和其他用混混淆。具体设置如表 1 所示。

表 1 用户信息表

名称	数据类型	大小	主键	空
ID	bson	12	Y	N
姓名	string	20	N	N
密码	string	20	N	N
性别	string	8	N	N

4.2 客户端设计

4.2.1 用户注册

客户端需对注册的信息进行简单的验证，如昵称为 5 个以上的字符或中文，如果昵称少于 5 个字符，则鼠标焦点回到错误的一栏，上方还有错误信息显示。注册界面如下图 7 所示。

用户注册

昵称输入不合法

昵称:

密码:

确认密码:

性别: ☒ 男 ☐ 女

图 7 用户注册界面

用户注册需在客户端进行一些格式限制，客户端处理代码如下：

```

$('#username').blur(function() {
    var name = $('#username').val();
    if(name === "" || !pattName.test(name)){
        $('#mess').html('昵称输入不合法');
        $('#username').focus().select();
    }
});

$('#password').blur(function() {
    var pass = $('#password').val();
    if(pass === "" || !pattPass.test(pass)){
        $('#mess').html('密码格式不正确');
        $('#password').focus().select();
    }
});

$('#password-repeat').blur(function() {
    var pass = $('#password').val();
    if($('#password-repeat').val() != pass){
        $('#mess').html('两次密码不一致');
        $('#password-repeat').focus().select();
    }
});

```

4.2.2 用户登录

用户如果没有填写昵称就将鼠标离开昵称一栏，页面上方会有红色标签提示昵称还没填写。界面如图 8 所示。

The image shows a user login window titled "用户登录" (User Login). It contains two input fields: "昵称:" (Nickname) and "密码:" (Password). The "昵称:" field has a placeholder text "请输入昵称" (Please enter nickname). The "密码:" field has a placeholder text "请输入密码" (Please enter password). Below the input fields is a blue button labeled "登录" (Login). The window has a close button in the top right corner.

图 8 用户登录界面

登录时进行简单校验，客户端用 jQuery 语言编写，核心代码如下：

```

$(function(){
    var $el = $('#dialog');
    $el.hDialog();
    $('#login').hDialog({modalHide: false});
    $('#login').hDialog({width:400,height:300});
    var $username = $('[name=username]');
    var $password = $('[name=password]');
    $username.blur(function() {
        if($username.val() === ""){
            $.tooltip('昵称还没填呢...');
            $username.focus();
        }
    });
    $password.blur(function() {
        if($password.val() == ""){
            $.tooltip('密码还没填呢...'); $password.focus();
        }
    });
});
});

```

在客户端会进行一个简单的校验，不成功便会在页面顶端显示错误信息，错误信息两秒后消失。如下图 9 所示。



昵称还没填呢...

图 9 登录失败信息图

4.2.3 用户聊天

为了使自己和对手的聊天内容区分，前台需将自己和对方的信息封装在不同的 id 中（HTML 文件），在 CSS 文件进行不同的美化。聊天是实时交互的，涉及到 WebSocket，前台代码如下：

```

iosocket.on('message', function(user,msg,roomid) {
    if(roomid === '<%=id%>'){
        $('#content').append($('#<li class="other"></li>')
        .html('<div>'+user+'</div><div class="text">'+msg+'</div>'));
    }
});

```

聊天效果如下图 10 所示：

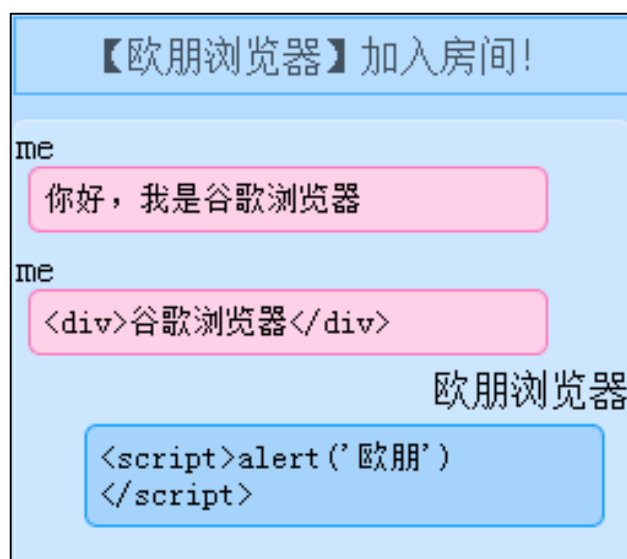


图 10 聊天效果图

4.2.4 音乐设置

HTML5 有自己的媒体音频 API，所以在这个功能上就非常简单，而且各浏览器支持良好。简单音乐播放功能有播放、暂停、音量调节和音乐跳转几个功能。一首音乐播放完以后，会自动跳到下一首循环播放，最后一首播放完跳到第一首重新播放。在这个功能上，自行添加的 CSS 样式会覆盖原有的音频界面，可以做到便捷且界面好看。核心代码如下：

```
$("#play").click(function() {
    if($(this).attr('class') === 'playControl'){
        music.play();
        $(this).html('暂停');
        $(this).attr('class','stopControl');
    }else{
        music.pause();
        $(this).html('播放');
        $(this).attr('class','playControl');
    }
});
```

音乐播放效果图如下图 11 所示。

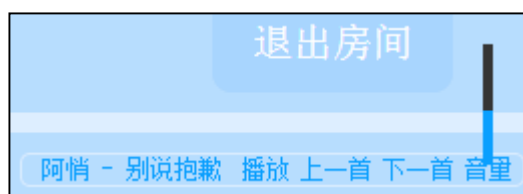


图 11 音乐播放效果图

4.3 服务器端设计

服务器最基本的功能是保证和客户端交互的实时性，既能实时接收客户端数据，又能实时向客户端发送数据。客户端与服务器端要实时交互的数据主要有注

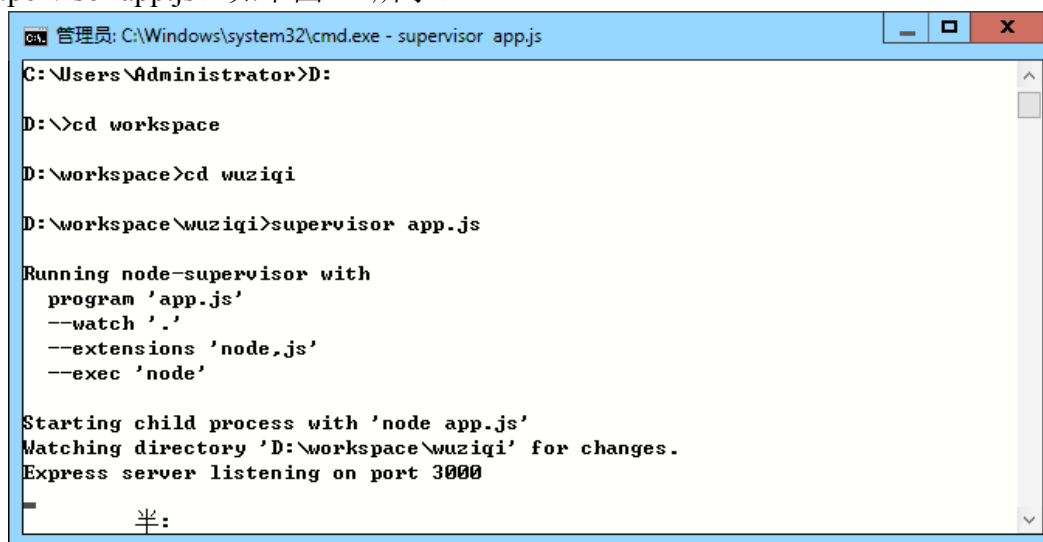
册数据、聊天数据和棋盘数据。注册数据发送到服务端后，服务端会保存客户端状态，以供聊天数据分发使用。聊天数据都是由用户在客户端输入，然后传到服务器，最后由服务器传到对应客户端，服务器只是起到一个转发功能，不用对数据的处理。棋盘数据是用户点击棋盘而发送到服务器的，服务器要进行相应的处理以判断是否一局游戏已结束，如果结束，则封装用户的输赢信息，再发送给客户端，客户端进行显示信息。以下为服务器端接收和发送数据部分的关键代码。

4.3.1 启动服务器和 socket

服务器语言采用 node.js，编写代码部分简单易懂，核心代码如下所示：

```
app.configure(function() {
  app.set('port', process.env.PORT || 3000);
  app.set('views', __dirname + '/views');
  app.set('view engine', 'ejs');
  app.set("view options", {
    layout: true
  });
  var server = http.createServer(app).listen(app.get('port'),
  function() {
    console.log("Express server listening on port "
+ app.get('port'));
  });
});
```

启动服务器只需在 cmd 命令中进入项目的目录，执行 node app.js 或者执行 supervisor app.js。如下图 12 所示。



```
管理员: C:\Windows\system32\cmd.exe - supervisor app.js
C:\Users\Administrator>D:
D:\>cd workspace
D:\workspace>cd wuziqi
D:\workspace\wuziqi>supervisor app.js

Running node-supervisor with
program 'app.js'
--watch '.'
--extensions 'node.js'
--exec 'node'

Starting child process with 'node app.js'
Watching directory 'D:\workspace\wuziqi' for changes.
Express server listening on port 3000
半:
```

图 12 服务器启动图

4.3.2 用户注册

前台在注册时已经进行了校验，但后台还需要进行校验。代码如下：

```

exports.doReg=function(req,res){
    User.find(newUser.name,function(err,user){
        if(user){
            req.session.error="该用户已经存在";
            return res.redirect("/reg");
        }
        newUser.save(function(err){
            if(err){
                req.session.error=err;
                return res.redirect("/reg");
            }
            req.session.user=newUser;
            req.session.success="注册成功";
            res.redirect("/roomlist.html");
        })
    })
}

```

用户注册失败后，会回到注册界面并且显示失败信息。如下图 13 所示。

该用户已经存在

图 13 用户注册失败图

4.3.3 用户登录处理

用户选择登录时，后台会对其数据与数据库进行校验，若信息不对，则会将错误信息封装在 session 中，在前台显示。

```

exports.doLogin=function(req,res){
    if(user.password!=password){
        req.session.error="用户密码错误";
        return res.redirect("/roomlist.html");
    }
    req.session.user=user;
    req.session.success="登录成功";
    res.redirect("/roomlist.html");
}

```

登录发生错误时，返回主页面并显示错误信息，如下图 14 所示。

用户密码错误

图 14 用户登录错误图

4.3.4 进入房间

用户登录之后便可选择一个可以进入的房间进入，如果房间后面显示“正在

战斗”，说明此时房间中已有两个人，无法第三人进入。如果用户直接通过连接进入“正在战斗”房间，将直接被退出。核心代码如下所示：

```
var i = 0;
for(var name in plays){
    if(plays[name] === username)
        i++;
}
if(i===0)
    fn(false);
else
    fn(true);
socket.broadcast.emit('user',plays,roomid,app.get(roomid+'fir'));
socket.broadcast.emit('check',roomid,false);
```

4.3.5 先手处理

每个先进入房间的用户为先手，先手的“申请先手”按钮是不可用状态，后手的“申请先手”按钮是可用状态。后手点击该按钮，先手同意后，便可互换。每次游戏完成一局以后，先手身份再次互换。核心代码如下所示：

```
socket.on('fir',function(roomid,setuser,username,fn) {
    if(setuser.length > 0)
        app.set(roomid+'fir',setuser);
    console.log('先手是： '+app.get(roomid+'fir'));
    var name = app.get(roomid+'fir');
    if(name === username)
        fn(true);
    var plays = app.get(roomid).split(' ');
    socket.broadcast.emit('user',plays,roomid,
        app.get(roomid+'fir'));
    socket.emit('user',plays,roomid,app.get(roomid+'fir'));
});
socket.on('applyfir',function(roomid,username) {
    socket.broadcast.emit('applyfir',roomid,username);
});
```

4.3.6 用户游戏

当先手没有点击“再来一盘”按钮时，无论谁点击棋盘都是无效的。游戏之前，双方的“认输”按钮是不可用状态。先手开始之后，“再来一盘”按钮失效，这时，双方“认输”按钮起效。五子棋对战游戏时，显示棋子和判断输赢都是实时性的，用户每下一个子，在后台都会进行判断，再将结果封装发送给用户。判断放在一个函数中，函数接收棋子位置后从该位置的横向、竖向、左上、左下四个方向进行判断是否五子连接。客户端根据接收到的信息性质进行显示，如果没有结束游

戏，客户端不做任何处理，双方继续游戏。如果在游戏中一方用户直接点击“认输”按钮，则会直接中断游戏，点击按钮的一方输棋。核心代码如下：

```
Socket.on('play',function(playid,roomid,map,player);
    var id = playid.split('_');
    var win = isEnd(parseInt(id[0]),parseInt(id[1]),player);
    socket.broadcast.emit('play',playid,roomid,win);
    socket.emit('winner',win);
});
```

游戏效果图如下 15 所示。

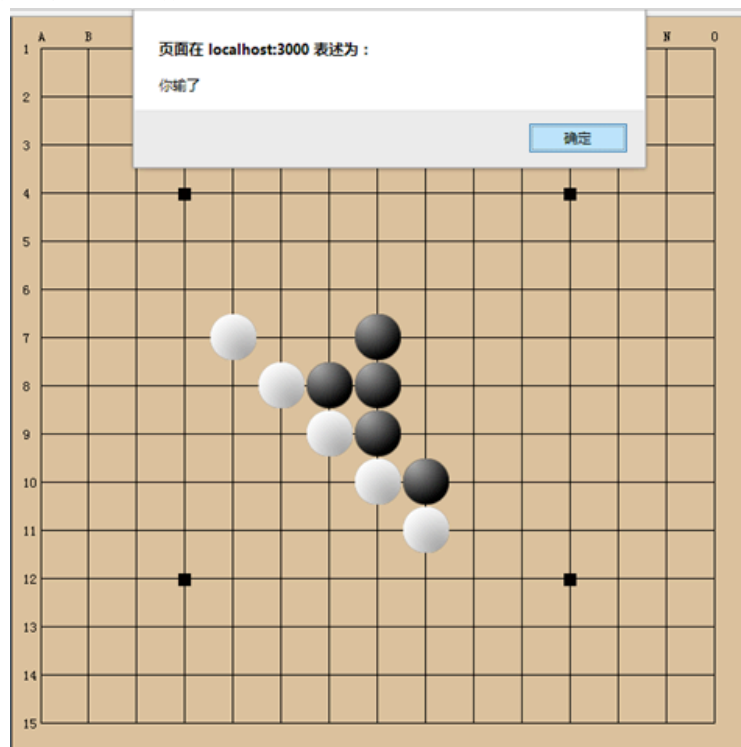


图 15 游戏效果图

4.3.7 退出房间

用户退出房间，则会将先手换给留在房间的用户，此房间对外也将显示“可以进入”，并且需要更新此房间的用户。核心代码如下：

```
socket.on('outroom',function(username,roomid,fn) {
    var plays = app.get(roomid).split(' ');
    var users = "";
    for(var i in plays){
        if(plays[i] != username){
            users += plays[i];
            console.log('users: '+i,users);
        }
    }
}
```



```

app.set(roomid,users);
    app.set(roomid+'fir',users);
    plays = app.get(roomid).split(' ');
    fn(true);

socket.broadcast.emit('user',plays,roomid,app.get(roomid+'fir'));
    socket.broadcast.emit('system message', '【'+username+'】退出房间!',roomid);
    });

```

4.4 界面设计

(1) 游戏大厅界面如下图 16 所示。



图 16 游戏大厅界面

(2) 游戏房间如下图 17 所示。



图 17 游戏房间图

5 总结

对于非常流行的 HTML5，相信很多人跟我一样对它非常着迷。在 HTML5 游戏开发领域，我掌握的很少，但是我对它的兴趣很浓厚。经过做毕业设计这段时间的努力，我对 HTML5 以及 Node.js 这门新学的语言有了更多的了解。此次设计，让我初步接触了 HTML 游戏，虽然我并没有用到 HTML5 游戏中非常流行的 Canvas 动画，但是我接触了 WebSoket 这一技术。不管怎么说，这种新兴的技术总在震惊着我们的大脑，让我们更加方便的写出自己想要的东西。

在做毕设的这段时间，用“痛并快乐着”来形容我一点也不为过。因为除了一些算法，我写的东西基本都是之前没有接触过的，或者说自学来的。很多时候，我都想放弃了，但是又不甘心。就这样，在同学的鼓励和自己的挣扎下，终于完成了设计。这让我对“编程语言”这个词有了更深的感情，也有了更多的信心去学习一门新语言，去迎接飞速更新的技术。无论将来信息怎样快速发展，语言怎样快速更换，只要我们有信心，就不会被淘汰，不会害怕困难。

致谢

本文是在指导老师赵靖老师的悉心指导下完成的，这让我看到了赵老师认真对待工作的态度，并激励我朝着这个方向发展，在我以后工作中一定能有所帮助。无论是平时的学习还是本次做毕业设计，赵靖老师都给予了我很多帮助和支持，让我受益匪浅。在此，我衷心的感谢赵老师对我的帮助！另外，同学们的鼓励也是一种巨大的支持，没有同学们的帮助，我也不一定这么快做好，所有，我还要感谢所有帮助过我的伙伴们，并且将这种互帮互助的精神传递下去！

参考文献

- [1] 龙奇.新一代网络技术标准 HTML5 的研究[J].科技信息,2011(10).
- [2] 刘再明.国产 HTML5 游戏路在何方?[J].互联网周刊,2013,12:22-24.
- [3] 刘再明.游戏:HTML5 商业化应用的急先锋[J].互联网周刊,2013(09).
- [4] 肖在昌,杨文晖,刘兵. 基于 WebSocket 的实时技术[J]. 电脑与电信,2012,12:40-42.
- [5] 温照松,易仁伟,姚寒冰. 基于 WebSocket 的实时 Web 应用解决方案[J]. 电脑知识与技术,2012,16:3826-3828.
- [6] 柳伯斯,阿伯斯,萨姆.HTML5 高级程序设计[M].北京:人民邮电出版社.2011: 3-65.
- [7] Jeanine Meyer[美].HTML5 游戏开发[M].北京: 人民邮电出版社, 2011:52-157.
- [8] Elizabeth Castro Bruce Hyslop[美].HTML5 与 CSS3 基础教程(第 7 版)[M].北京: 人民邮电出版社, 2012:1-3.
- [9] Makzan[美].HTML5 游戏开发实战[M].北京: 人民邮电出版社, 2011:10-43.
- [10] 香卡 (Shankar,A.R.)[印].HTML5 游戏开发进阶指南[M].北京: 电子工业出版社, 2013:1-3.

Design and Implementation of the HTML5 Web-Based Interactive Game

Student majoring in computer science and technology Dong Yuanyuan

Tutor ZHAO Jing

Abstract: HTML5 specification developed from the popular to this process, people use it to develop the game gradually increased, and some even predicted that it will replace flash. HTML5 field advantage in the game is the ability to achieve cross-platform game coding transplantation; in addition, HTML5 support WebSocket API, which also makes HTML5 game development has become, especially in the important trend of mobile gaming platform. This article will introduce a web version of the backgammon game, the main features include online chat, listen to music and play games, users do not need to download APP, you can open a browser game. In this paper, Node.js as a development environment, using mongodb create a simple database, web landscaping use HTML5 + CSS3 + jQuery, real-time information exchange via WebSocket implementation, simple, simple and beautiful interface.

Keywords: HTML5; Node.js; WebSocket; real-time; interactive games;