

密级：公开

基于 HTML5 与 Nodejs 开发实时性网 页对战游戏坦克大战

The Development of Battle City Web Page Game Based on HTML5 and Nodejs

学 院：信息科学与工程学院

专 业 班 级：计算机科学与技术 0703 班

学 号：070405066

学 生 姓 名：王潇

指 导 教 师：无

2011年06月

摘 要

随着互联网技术的不断发展和计算机性能的不断提升，用户享受互联网的方式也开始不断的变化，越来越多的应用可以直接以浏览器打开的方式使用，数据则是存储在云端，更加的安全可靠，方便用户共享数据。在各种应用不断的趋于更加网络化的同时，一种新的技术标准也随之产生，那就是 HTML5（HTML 是一种标记语言也是一种标准，互联网开发居于此）。HTML5 是近十年来 Web（互联网）标准最巨大的飞跃，和以前的版本不同，HTML5 并非仅仅用来表示 Web 内容，它的使命是将 Web 带入一个成熟的应用平台，在这个平台上，视频、音频、图象和动画，以及同电脑的交互都被标准化。也就是说 HTML5 是通过开放的技术和标准来实现一个无插件模式的富客户端，这个客户端可以在移动终端、PC（个人计算机）或者其他可以有网络的地方轻松地运行。所以此时研究 HTML5，并且大胆的实践有着重要的意义。

课题内容是使用 HTML5 技术基于 Web 的环境来制作一个网络版的“坦克大战”游戏应用，该游戏使用了 HTML5 中的 Canvas（画布）、WebSocket（HTML5 中最新使用的一种网络连接协议，用于实时通讯）、Audio（HTML5 中控制声音的技术）等最新的技术。目前 HTML5 标准尚未成熟，很多地方仍在变动，API（应用程序接口）仍不是很稳定，所以还无法真正的完全商用或者全部普及开，但这并不会影响我们来研究它。服务器端采用的 Nodejs（一种 Javascript 程序语言框架，可以在服务器端运行 Javascript 语言）这种新型的框架来开发，Nodejs 使用 Javascript 语言来开发，使整个应用从客户端到服务器端编程都是使用 Javascript，大大的减少学习成本和开发成本，并且 Nodejs 性能强大，有很大的研究价值。

该游戏服务器运行后，用户可以直接通过浏览器打开，实时的同其他玩家在游戏中对战，简单高效并且有声音和实时性的数据传输。用户可以用键盘控制自己的坦克，在地图中和自己的队友配合去攻击另一队的坦克来获取胜利，整个游戏趣味性很强。

关键词：HTML5；互联网实时性通信；Nodejs；网络游戏

Abstract

With the increasing development of the Internet technology and the constantly rising of the computer performance, users also begin to enjoy the Internet in the constantly changed ways. More and more application can be directly use in the way of opening the browser, data are stored in the clouds, and therefore it's more safe and reliable and continent for users to share data. In the same time of various application continuously tends to be more webified, a new standard of technology is produced, which is HTML5. HTML5 is the most giant leap of Web standards in nearly 10 years and it is different from the previous version. HTML5 is not only used to express the content of the Web, meanwhile, its mission is to bring the Web into a mature application platform, on which the video, audio, image and animation, and the interaction of computers are all standardized. That is to say that through the opening technology and standards HTML5 achieves a no plug-ins model of client. The client can be operated in the mobile terminal, PC or other places which have the network. Therefore, it has significance to study and boldly practice the HTML5.

The contents of the subject topics is to use the HTML5 create a Web edition of "tank war" game application based on the environment of Web. The game operates the Canvas, WebSocket, Audio and some latest technology. At present, the standards of HTML5 have not yet mature and some points are still in the variation and API still is not very stable, so it can't be completely commercial or in a good popularization. However, we can still have a good study of it. The server is used Nodejs, a new Javascript framework, to develop. Nodejs is used Javascript to develop, which makes whole application, from the client to the server, using Javascript to program, greatly reduce the cost of learning and development costs. Meanwhile, Nodejs has the strong function and have the great value to study.

After the operation of the game, users can directly open it through the browser and against with other players, with a simple, effective, voiced and data transmission of real-time. Users can use the keyboard to control their tanks and to attack another team to win the game with his teammates in the map. The whole game has a strong sense of interesting.

Keywords: HTML5; internet real-time communications; Nodejs; online games

目 录

摘 要.....	I
Abstract.....	II
第 1 章 引言.....	1
1.1 课题的目的和意义.....	1
1.2 国内外发展现状.....	2
1.3 课题主要的内容和功能介绍.....	4
1.3.1 课题主要的内容.....	4
1.3.2 坦克大战游戏功能介绍.....	5
第 2 章 相关技术及开发方法简介.....	6
2.1 相关技术简介.....	6
2.1.1 HTML5 标准简述及意义.....	6
2.1.2 网络实时性技术及 WebSocket.....	7
2.1.3 Nodejs 简介.....	11
2.2 开发工具及运行环境.....	13
2.2.1 Web 开发工具和 Aptana 简述.....	13
2.2.2 浏览器发展和 Chrome 浏览器介绍.....	14
第 3 章 游戏玩法设计和需求分析.....	17
3.1 游戏玩法设计.....	17
3.2 需求分析.....	17
3.2.1 系统角色.....	17
3.2.1 用例分析.....	17
第 4 章 总体设计及程序功能分析.....	18
4.1 程序总体功能设计.....	18
4.1.1 程序功能模块分析.....	18
4.1.2 地图读取及共享分析.....	19
4.1.3 声音的控制分析.....	19
4.1.4 联网对战分析.....	19
4.2 程序流程.....	20
4.3 模块及对应的方法.....	22
第 5 章 详细设计.....	26
5.1 服务器端设计.....	26
5.1.1 服务器端的运行环境.....	26
5.1.2 服务器端的设计.....	26

5.1.3 服务器端部分代码.....	27
5.2 地图功能模块.....	28
5.2.1 读取地图.....	28
5.2.2 绘制地图.....	28
5.2.2 地图模块部分代码.....	29
5.3 坦克模块的设计.....	31
5.3.1 读取服务器端信息.....	31
5.3.2 控制坦克.....	31
5.3.3 绘制画面.....	31
5.3.4 坦克模块的部分实现代码.....	31
5.4 炮弹模块的功能.....	35
5.4.1 读取服务器端信息.....	35
5.4.2 碰撞检测.....	35
5.4.3 击毁功能.....	35
5.4.4 绘制画面.....	35
5.5 客户端与服务器端通信.....	35
第 6 章 结论.....	37
参 考 文 献.....	38
致 谢.....	40

第 1 章 引言

1.1 课题的目的和意义

随着互联网的不断发展，互联网对人们的生活也在不断地变化和加强，人们渐渐地已经开始习惯于互联网所带来的各类服务与应用，并且享受着它带来的便利和丰富。当然随着互联网的基础用户和开发者不断增多，人们不断地去创造和完善它的速度也越来越快，无论从技术层面还是商业层面，新的模式和方法层出不穷，整个行业新陈代谢不断加速，新的应用和新的公司不断地挑战着新的技术和新的模式，由于整个行业存在创业成本很低、技术成本门槛却较高等特点，大公司也不断地推动着新的技术，基于此来创建一定的技术壁垒，同时基于此建立全新的服务和技术。但是毕竟整个互联网是一个开放的环境，大家需要一种标准，而不是一家独大的竞争下去，这样就一定会失去民心，所以 HTML5 这种更加新的标准和技术体系在不断地革新中发展壮大。

HTML5 是近十年来 Web 标准最巨大的飞跃，和以前的版本不同，HTML5 并非仅仅用来表示 Web 内容，它的使命是将 Web 带入一个成熟的应用平台，在这个平台上，视频、音频、图象和动画，以及同电脑的交互都被标准化。尽管 HTML5 的实现还有很长的路要走，但 HTML5 正在改变 Web。另外，目前的互联网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，如物理位置定位信息等新的数据采集和多点触控、重力感应等新的交互方式的应用，未来必将也会改变人们生活的方方面面。

HTML5 中也加入了以上各类新的元素和功能，同时也考虑到了非健全人事使用互联网时可能带来的问题^[1]。所以，研究 HTML5 这种技术体系来尝试做一些应用是一个千载难逢的机遇，而基于这个变革的年代去使用类似的技术来创造更多新的应用更是能够改变这个世界的一次机会。

该课题是基于 HTML5 来做一个最佳实践，完成之前不可能使用 WEB 技术来完成的功能，性能照比之前的技术体系要高出很多，而且有足够的前瞻性和创新性。互联网和其他的行业最大的区别就是这是一个创新驱动并且快速变化的行业，一种新的技术可能就会创造出新的应用、新的商业模式，该课题是基于 HTML5 技术在 Web 环境中实现的一个实时性的联网游戏，通过使用 HTML5

中的 WebSocket 来实现实时性，这无疑是对最新技术的一次最佳实践和学习，也是对自己的能力的一次历练。一个联网的游戏最难解决的问题无疑就是实时性的问题，同步数据的数量、何时同步数据，这些都是需要认真思考和设计的，不仅仅是简单的程序设计，而是复杂度较高真正的软件工程。

游戏软件是计算机诞生之后就有的产物，最开始是一些计算机极客们无聊时写着玩的代码，渐渐地发展成为了一个巨大的产业，游戏软件由最开始的单机版游戏逐渐发展为大型的网游，但是随着整个行业新生模式的诞生，尤其是如 SNS、微博等新型的以人与人关系类的网络，游戏也在向着小型互动的 SNS 型转变，这样的游戏有着互动性强，直接在网页中运行，用户体验较简单容易，就如之前风靡一时的“偷菜”类游戏。游戏可以带给人们快乐，可以使人们享受与人配合交流的快乐，也可以是打破尴尬的一种方式。对于程序员而言能够编写一个游戏程序带给用户快乐，这无疑也会带给自己成就感并且是一件很有意义和价值的事情。

1.2 国内外发展现状

HTML 的发展历经了多个年代，但时间其实不过也在 20 年左右。1993 年 HTML 首次以因特网草案的形式发布。20 世纪 90 年代的人见证了 HTML 的大幅发展，从 2.0 版，到 3.2 版和 4.0 版，再到 1999 年的 4.01 版。随着 HTML 的发展，W3C（万维网联盟）掌握了对 HTML 规范的控制权。然而在快速发布了这四个版本之后，业界普遍认为 HTML 已经到了穷途末路，对 Web 标准的焦点也开始转移到了 XML（一种数据存储的形式）和 XHTML（XHTML 类似 HTML 的一种标记语言，也同样是 Web 开发的一个标准，要求页面结构像 XML 一样严谨）上，HTML 被放到了次要的位置。不过在此期间，HTML 体现了顽强的生命力，主要的网站内容还是基于 HTML 的。为能支持新的 Web 应用，同时克服现有的缺点，HTML 迫切需要添加新的功能，制定新的规范。

致力于将 Web 平台提升到一个新的高度，一小组人在 2004 年成立了 WHATWG（Web 超文本应用技术工作组）。他们创立了 HTML5 规范，同时开始专门针对 Web 应用开发新的功能，这被 WHATWG 认为是 HTML 中最薄弱的环节。Web2.0 实至名归，开创了 Web 的第二时代^[2]。旧的静态网站逐渐让位于需要更多特性的动态网站和社交网站，这些网站中的新功能数不胜数。2006 年 W3C 又重新介入 HTML，并于 2008 年发布了 HTML5 的工作草案。2009 年，

XHTML2 工作组停止工作。2010 年因为 HTML5 能解决非常实际的问题，所以在规范还未定稿的情况下，各大浏览器厂家就已经按耐不住了，开始对旗下产品进行升级以支持 HTML5 的新功能。得益于浏览器的实验性反馈，HTML5 规范也得到了持续的完善，HTML5 以这种方式迅速融入到了对 Web 平台的实质性改进中。到目前为止 HTML5 规范已经以工作草案的形式发布了，但还不是最终版本，不过现在在移动终端和 Apple 公司为主的产品的推动下，HTML5 已经得到了广泛的使用^[3]。

HTML5 草案的前身名为 Web Applications 1.0，是在 2004 年由 WHATWG 提出，再于 2007 年获 W3C 接纳，并成立了新的 HTML 工作团队。在 2008 年 1 月 22 日，第一份正式草案发布。WHATWG 表示该规范是目前仍在进行的工作，仍须多年的努力。目前 Firefox、Chrome、Opera、Safari（版本 4 以上）及 Internet Explorer 9 已支持 HTML5 技术（Firefox、Chrome、Opera、Safari 和 Internet Explorer 9 均为浏览器品牌）。尽管 HTML5 已经在网络开发人员中非常出名了，但是它成为主流媒体的一个话题还是在 2010 年的 4 月，当时 Apple 公司的 CEO 乔布斯发表一篇题为“对 flash 的思考”的文章，指出随着 HTML5 的发展，观看视频或其它内容时，Adobe Flash 将不再是必须的。这引发了开发人员间的争论，包括 HTML5 虽然提供了增强的功能，但开发人员必须考虑到不同浏览器对标准不同部分的支持程度的不同，以及 HTML5 和 Flash 间的功能差异。HTML 5 的标准草案目前已进入 W3C 制定标准 5 大程序的第 1 步。负责编纂标准格式文件的 Google 公司的代表 Ian Hickson 预期，可能得等到 2012 年才会推出建议候选版（W3C Candidate Recommendation）^[4]。

据美国知名科技博客网站 TechCrunch 报道，Facebook（一家美国的社交网站，现在被誉为全世界最火的网站）已秘密制定了一项名为“斯巴达”（Project Spartan）的移动平台服务计划，此举意在以全新方式向苹果 iOS 平台（苹果公司的移动终端操作系统）和 Google 的 Android 平台（Google 公司的移动终端操作系统）以及相关移动服务发起强有力的挑战。“斯巴达”目前仍然属于 Facebook 秘密开发的产品，目前确切可知的消息是“斯巴达”是一款基于 HTML5 技术标准的移动服务平台，目标群目前锁定为使用移动版 Safari 浏览器的 iPhone 手机和 iPad 用户。目前已知的“斯巴达”平台的工作原理是：该平台基于 HTML5 技术，而 Safari 浏览器又对 HTML5 有比较好的支持，用户通过 Safari 访问“斯巴达”平台后，不仅能够在该平台使用 Facebook 的基本服务，还能够获取使用由

外部开发者基于该平台开发的相关应用程序。而且微软的说法最新的 Windows 8（微软的下一代操作系统）将会支持两种应用，其一是传统桌面应用程序，另外就是 HTML5 网络应用。Windows 8 中通过 Internet Explorer 10（微软下一代浏览器）作为其 HTML5 应用核心，所有现有的 Windows 程序将支持 ARM（ARM 是一种处理器架构，也是一种处理芯片）片上系统架构。微软称将给开发者提供 Windows 8 HTML 和 JavaScript 应用开发的 API 和 SDK（Software Development Kit，即软件开发工具包）。

HTML5 无疑是客户端的发展的一个巅峰，但是服务器端的变化也是显著的，从之前的 CGI（Common Gateway Interface，通用网关接口，在物理上是一段程序，运行在服务器上，提供同客户端 HTML 页面的接口）到 JAVA、J2EE（一种商用的 JAVA 开发框架，更是一种软件生产标准），再到 LAMP（Linux+Apache+Mysql+Perl/PHP/Python 一组常用来搭建动态网站或者服务器的开源软件，本身都是各自独立的程序，但是因为常被放在一起使用，拥有了越来越高的兼容度，共同组成了一个强大的 Web 应用程序平台）和 Ruby on Rails（Ruby on Rails 是一个可以使你开发，部署，维护 Web 应用程序变得简单的框架，基于 Ruby 语言），随着 HTML5 的兴起下一个服务器端架构将会诞生，再次改变技术圈的架构。本课题使用了较为有前景的一个全新的架构 Nodejs，Nodejs 是一个 Javascript 运行环境。实际上它是对 Google V8 引擎进行了封装（Google V8 引擎是用来解析 Javascript 的专用运行环境，以速度快著称）。V8 引擎执行 Javascript 的速度非常快，性能非常好。Nodejs 对一些特殊用例进行了优化，提供了替代的 API，使得 V8 在非浏览器环境下运行得更好。Nodejs 也是刚刚兴起的一种服务器端的开发架构，使得 Javascript 能够在服务器端运行，而且由于 Javascript 的函数式编程等特性，基于非阻塞式的编程更加的有优势。国内研究比较超前的是淘宝的 CNodejs 团队，他们做了大量测试、实践和分享。

1.3 课题主要的内容和功能介绍

1.3.1 课题主要的内容

课题主要的内容包括：

- (1) 使用 HTML5 Canvas 来开发动画效果。
- (2) 基于 HTML5 WebSocket 的实时数据传输。
- (3) 服务器端 Nodejs 的使用。

-
- (4) 基于 HTML5 Audio 来实现控制声音。
 - (5) 网络实时性游戏的设计和实现。
 - (6) 地图的绘制与实时性地图的控制。

1.3.2 坦克大战游戏功能介绍

玩家打开浏览器，输入地址即可进入坦克大战游戏并进入联机状态，联机后将会被自动分配到黄方或者绿方，玩家可以通过键盘控制自己的坦克，黄、绿双方各自将会以消灭对方为己任，互相攻击。游戏可以负载多人，地图可以在服务器端通过修改数组改变。

第 2 章 相关技术及开发方法简介

2.1 相关技术简介

2.1.1 HTML5 标准简述及意义

HTML5 是 HTML 下一个的主要修订版本，现在仍处于发展阶段。目标是取代 1999 年所定订的 HTML 4.01 和 XHTML 1.0 标准，以期能在互联网应用迅速发展的时候，使网络标准达到符合当代的网络需求。广义论及 HTML5 时，实际指的是包括 HTML、CSS 和 JavaScript 在内的一套技术组合。它希望能够减少浏览器对于需要插件的丰富性网络应用服务（plug-in-based rich internet application, RIA），如 Adobe Flash（Adobe 公司的一种技术体系，使用 Actionscript 语言，开发后生成的程序以插件形式在浏览器端运行）、Microsoft Silverlight（微软公司的一种技术，同样是以插件形式在浏览器端运行），与 Oracle JavaFX（基于 JAVA 的一种技术，SUN 公司被 Oracles 收购后归于 Oracle 公司，以插件形式在浏览器端运行）的需求，并且提供更多能有效增强网络应用的标准集。

HTML5 添加了许多新的语法特征，其中包括<video>，<audio>，和<canvas>元素（<video>，<audio>，和<canvas>均为 HTML5 中的新标记），同时集成了 SVG（HTML5 中新加入的矢量图形实现方式）内容。这些元素是为了更容易的在网页中添加和处理多媒体和图片内容而添加的。其它新的元素包括<section>，<article>，<header>和<nav>（<section>，<article>，<header>和<nav>均为 HTML5 新添加的标记），是为了丰富文档的数据内容。新的属性的添加也是为了同样的目的。同时也有一些属性和元素被移除掉了。一些元素，像<a>，<cite>和<menu>（<a>，<cite>和<menu>均为 HTML 标签）被修改，重新定义或标准化了。同时 APIs 和 DOM（Document Object Model，文档对象模型）已经成为 HTML5 中的基础部分了。HTML5 还定义了处理非法文档的具体细节，使得所有浏览器和客户端程序能够一致地处理语法错误。

HTML5 提供了一些新的元素和属性，反映典型的现代用法网站。其中有些是技术上类似<div>和标签，但有一定含义，例如<nav>（网站导航块）和<footer>。这种标签将有利于搜索引擎的索引整理、小屏幕设备和视障人士使用。同时为其他浏览要素提供了新的功能，通过一个标准接口，如<audio>和

<video>标记。一些过时的 HTML4 标记将取消，其中包括纯粹用作显示效果的标记，如和<center>（HTML 中为数不多的具有修改样式属性的标签），因为它们已经被 CSS 取代。还有一些通过 DOM 的网络行为^[5]。

尽管和 SGML（Standard Generalized Markup Language,标准通用标记语言）在标记上的相似性，HTML5 的句法并不再基于它了，而是被设计成向后兼容对老版本的 HTML 的解析。它有一个新的开始行看起来就像 SGML 的文档类型声明，<!DOCTYPE html>（<!DOCTYPE>声明位于文档中的最前面的位置，处于<html>标签之前，此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范）这会触发和标准兼容的渲染模式。在 2009 年 1 月 5 号，HTML5 添加了 Web Form 2.0（处理表单的功能）的内容。

HTML5 还提供了许多新的应用程序接口（API），除了原先的 DOM 接口，HTML5 增加了更多样化的 API：

(1) 实时二维绘图。Canvas API：有关动态产出与渲染图形、图表、图像和动画的 API。

(2) 定时媒体播放 HTML5 音频与视频：HTML5 里新增的元素，它们为开发者提供了一套通用的、集成的、脚本式的处理音频与视频的 API，而无需安装任何插件。

(3) 通信网络 Communication APIs：构建实时和跨源（cross-origin）通信的两大基础：跨文档通信（Cross Document Messaging）与 XMLHttpRequest Level 2。

(4) Geolocation API：用户可共享地理位置，并在 Web 应用的协助下享用位置感知服务（location-aware services）。

(5) Web SQL Database，一个本地的 SQL 数据库。

(6) WebSocket，一个基于浏览器的 socket（一种网络协议，支持所有的互联网通信），支持实时性的数据传输。

(7) 其他：离线存储数据库（离线网络应用程序）、编辑、拖放等。

2.1.2 网络实时性技术及 WebSocket

正常情况下，浏览器访问 Web 页面时，一般会向页面所在的 Web 服务器发送一个 HTTP 请求。Web 服务器识别请求，然后返回响应。大多数情况下，如股票价格、新闻报道、余额查询、交通状况、医疗设备读取数据等，当内容

呈现在浏览器页面上时，可能已经没有了时效性。如果用户想要获得最新的实时信息，就需要不断地手动刷新页面，这显然不是一个明智的做法。目前实时 Web 应用的实现方式，大部分是围绕轮询和其他服务器端推送技术展开的，其中最著名的是 Comet。Comet 技术可以让服务器端主动以异步方式向客户端推送数据，它会使针对传输消息到客户端的响应延迟完成^[6]。

(1) 使用轮询。浏览器会定期发送 HTTP 请求，并随即接收响应，请求过程如图 2-1。

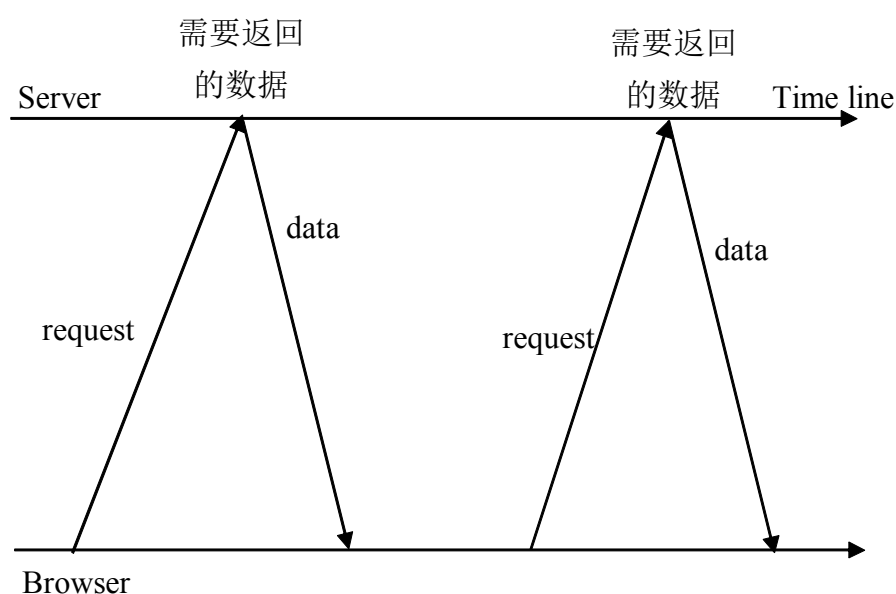


图 2-1 传统网络连接通过轮询方式获取数据的过程图

这项技术是浏览器在实时信息传送方面的首次尝试。显然，如果知道消息传递的准确时间间隔，轮询将是一个很好的办法，因为可以将客户端的请求同步为只有服务器上的信息可用时才发出。但是，实时数据往往不可预测，不可避免会产生一些不必要的请求，在低消息率情况下会有很多无用的连接不断地打开和关闭^[7]。

(2) 使用长连接。浏览器向服务器发送一个请求，服务器会在一段时间内将其保持在打开状态。如果服务器在此期间收到一个通知，就会向客户端发送一个包含消息的响应。如果时间已到却还没有收到通知，服务器会发送一个响应消息来终止打开的请求。然而，最关键的是，当信息量很大时，与传统轮询

方式相比，长连接方式并无实质上的性能改善^[8]。

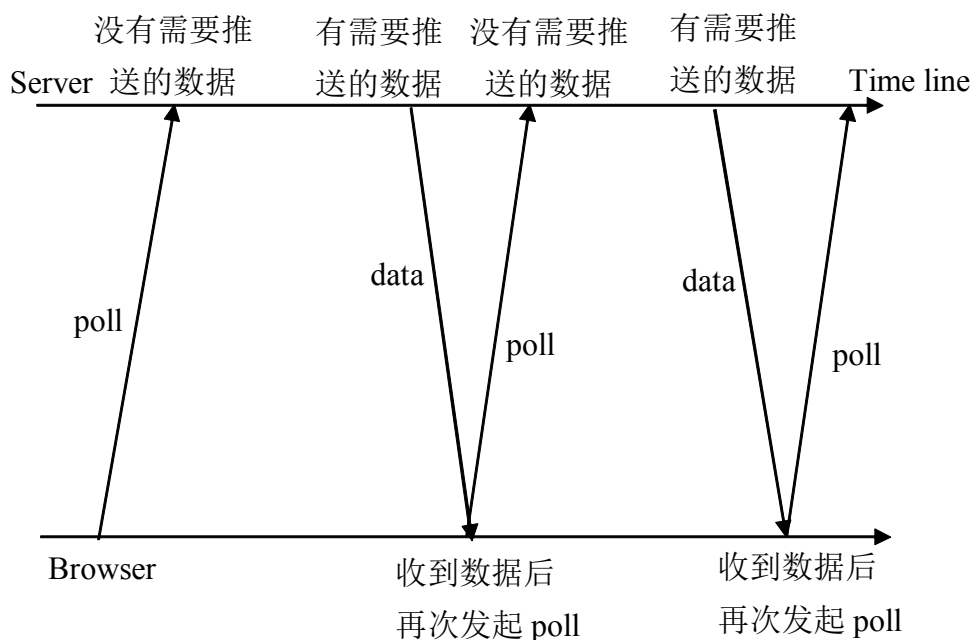


图 2-2 传统网络连接通过长连接方式获取数据的过程图

(3) 使用流解决方案。浏览器会发送一个完整的 HTTP 请求，但服务器会发送并保持一个处于打开状态的响应，该响应持续更新并无限期或是在一段时间内处于打开状态。每当有消息可发送时，该响应就会被更新，但服务器永远不会发出响应完成的信号，这样连接就会一直保持在打开状态以便后续消息的发送。但是，由于流仍是封装在 HTTP 中，其间的防火墙和代理服务器可能会对响应消息进行缓冲，造成消息传递的延时。因此，当检测到缓冲代理服务器时，许多流解决方案就回退到长连接方式。此外，可利用 TLS、SSL 连接来保护响应不被缓冲，但在这种情况下，每个连接的创建和清除会消耗更多的服务器资源^[9]。

综上所述，所有这些提供实时数据的方式都会涉及 HTTP 请求和响应报头，其中包含有大量额外的、不必要的报头数据，会造成传输延迟。最重要的是，全双工连接需要的不仅仅是服务器到客户端的下行连接。为了在半双工 HTTP 的基础上模拟全双工通信，目前的许多解决方案都使用了两个连接：一个用于

下行数据流，另一个用于上行数据流。这两个连接的保持和协作也会造成大量的资源消耗，并增加了复杂度。简而言之，HTTP 技术并不是为了实现实时全双工通信设计的。当开发人员试图对上述方案继续扩展时，情况会变得更糟。模拟基于 HTTP 的双向浏览器通信是非常复杂和易错的，而且复杂度不可控。虽然最终用户感觉 WEB 应用像是实时的，但是这种“实时”体验的代价非常高，包括额外的时间延迟、不必要的网络流量和 CPU 性能消耗^[10]。

WebSocket 实现了通过浏览器直接于服务器形成全双工连接。WebSocket 是 HTML5 一种新的协议，它实现了真正的浏览器与服务器全双工通信 (full-duplex)。WebSocket 请求过程如图 2-3。

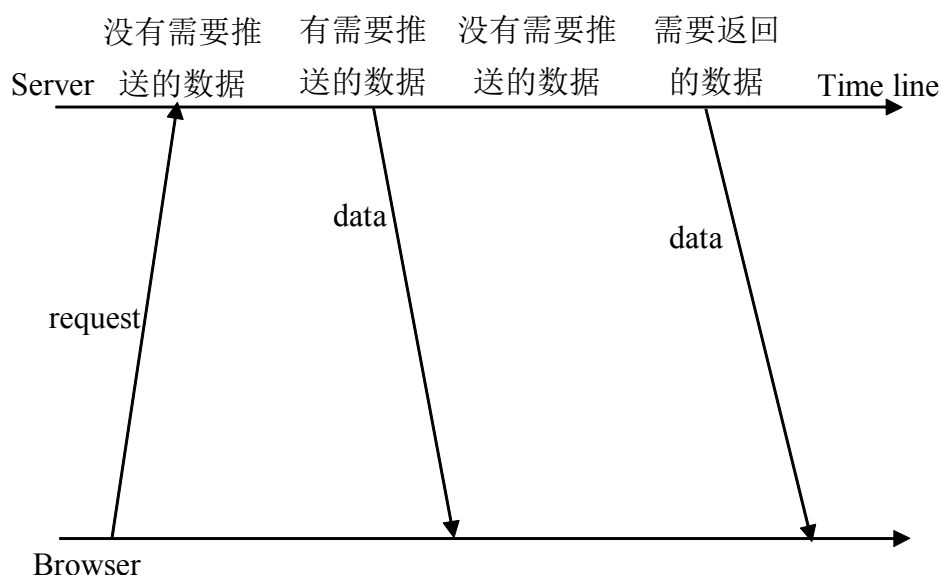


图 2-3 网络连接通过 WebSocket 方式获取数据的过程图

WebSocket 是 HTTP 协议的扩展，用 HTTP 握手之后，服务器和浏览器就使用这条 HTTP 链接下的 TCP 连接来直接传输数据，抛弃了复杂的 HTTP 头部和格式，WebSocket 的格式是基于帧 (Frame) 的，最小的帧只有 2 个字节。在文本帧中，每一帧始于 0x00 直接，止于 0xFF 字节，数据使用 UTF-8 编码。与 long polling 相比，通信的次数虽然一样多，但是 Web Socket 的帧头的标记部分也比 long polling 的 request 的 header 部分简洁得多 (几个字节与几 K 字节的差别)。为了建立 WebSocket 通信，客户端和服务器在初始握手时，将 HTTP 协

议升级到了 WebSocket 协议，一旦连接建立成功，就可以在全双工模式下在客户端和服务器之间来回传送 WebSocket 消息。这就意味着，在同一时间、任何方向，都可以全双工发送基于文本的消息^[11]。

2.1.3 Nodejs 简介

Nodejs 是一个 Javascript 运行环境。实际上它是对 Google V8 引擎进行了封装。V8 引擎执行 Javascript 的速度非常快，性能非常好。Nodejs 对一些特殊用例进行了优化，提供了替代的 API，使得 V8 在非浏览器环境下运行得更好。例如，在服务器环境中，处理二进制数据通常是必不可少的，但 Javascript 对此支持不足，因此，V8.Node 增加了 Buffer 类，方便并且高效地处理二进制数据。因此，Nodejs 不仅仅简单的使用了 V8，还对其进行了优化，使其在各环境下更加实用^[12]。Nodejs 的运行架构如图：

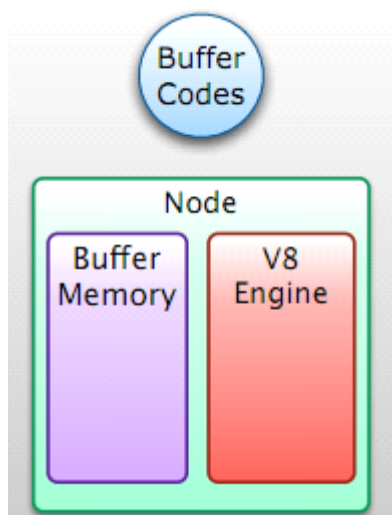


图 2-4 Nodejs 的运行架构图

V8 引擎本身使用了一些最新的编译技术。这使得用 Javascript 这类高级语言编写出来的代码与用 C 这类低级语言写出来的代码性能相差无几，却节省了开发成本。对性能的苛求是 Nodejs 的一个关键因素。Javascript 是一个事件驱动语言，Nodejs 利用了这个优点，编写出可扩展性高的服务器。Nodejs 采用了一个称为“事件循环（event loop）”的架构，使得编写可扩展性高的服务器变得既容易又安全。提高服务器性能的技巧有多种多样，Nodejs 选择了一种既能提

高性能，又能减低开发复杂度的架构。这是一个非常重要的特性，并发编程通常很复杂且布满地雷，Node 绕过了这些，但仍提供很好的性能^[13]。

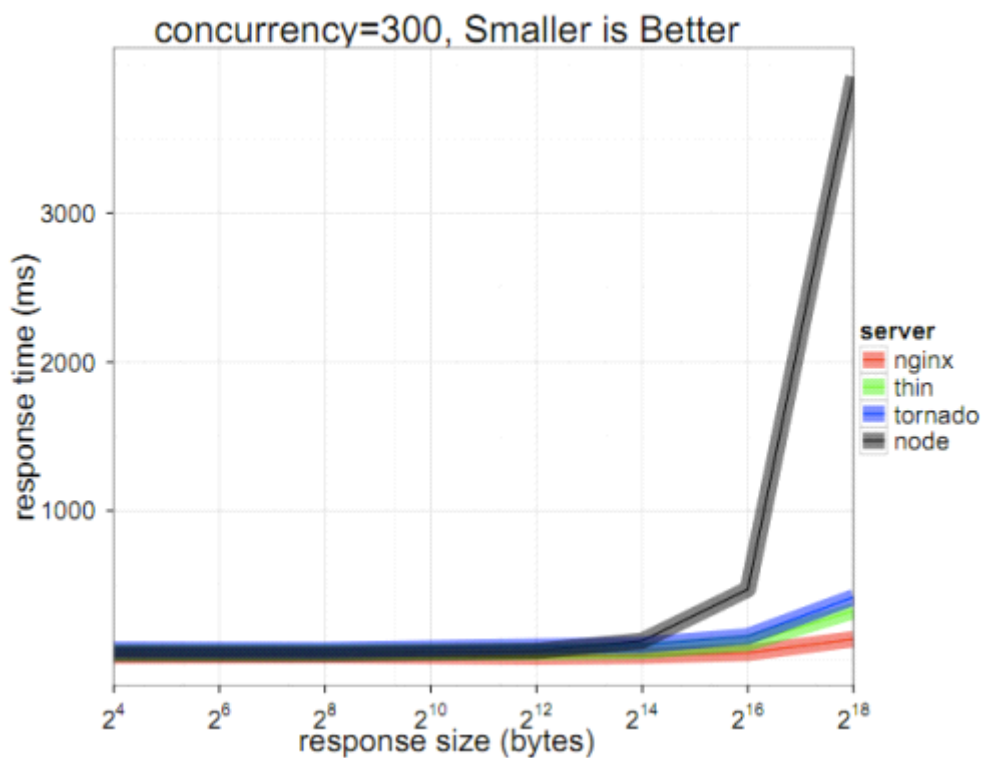


图 2-5 Nodejs 在处理数据时带来性能提升的比例图

Nodejs 采用一系列“非阻塞”库来支持事件循环的方式。本质上就是为文件系统、数据库之类的资源提供接口。当你向文件系统发送一个请求时，无需等待硬盘寻址并检索文件，硬盘准备好的时候非阻塞接口会通知 Nodejs。该模型以可扩展的方式简化了对慢资源的访问，直观，易懂。尤其是对于熟悉 onmouseover、onclick 等 DOM 事件的用户，更有一种似曾相识的感觉。虽然让 Javascript 运行于服务器端不是 Node 的独特之处，但却是其一强大功能。不得不承认，浏览器环境限制了我们选择编程语言的自由。任何服务器与日益复杂的浏览器客户端应用程序间共享代码的愿望只能通过 Javascript 来实现。虽然还存在其他一些支持 Javascript 在服务器端运行的平台，但因为上述特性，Node 发展迅猛，成为事实上的平台。

2.2 开发工具及运行环境

2.2.1 Web 开发工具和 Aptana 简述

由于 Web 开发所作出的程序需要在 Web 浏览器上面来观看效果，所以调试和编译运行等过程主要由浏览器或者后端服务器来完成，这样使得 Web 开发工具的主要功能其实就是一个编辑器。Web 开发工具渐渐的分两大类，一类是基于编辑器的，如 Editplus、Vim 等，另一个则是由 Eclipse 构建的^[14]。

编辑器类以 Editplus 为例，主要是对编程过程中的代码高亮显示，而且可以提供自定义模板等功能，提高编程效率。而且 Editplus 打开速度很快，功能强大，很多 Web 工程师喜欢^[15]。

Eclipse 则是由 IBM 研发的大型集成化编译环境（IDE），原本主要用于编写大型的 Java 程序，后来 IBM 将其变为开源项目，而且 Eclipse 以插件的机制可扩展，所以强大的扩展性，使其定制灵活，所以后续有很多基于 Eclipse 开发的 IDE。如编写 PHP 的 ZendStudio、编写 Actionscript 的 FlashDevelop 等。虽然 Eclipse 功能强大，但是打开速度较慢，系统内存消耗较大。

Aptana 正是基于 Eclipse 开发的一款强大的 IDE，对 HTML5 和 Javascript 有强大的支持，很好的代码高亮显示和代码提示功能，且运行稳定、扩展性强。Aptana 不仅仅支持 HTML5 和 Javascript，同时也支持 Ruby、Python 等 Web 编程语言。其最广为人知的是它非常强悍的 Javascript 编辑器和调试器，去年 Aptana 吸收了 Radrails 项目，添加了非常强大的 Ruby on Rails 支持。但是 Aptana 前进的脚步并未停止。随着苹果公司 iPhone 手机的发布，Aptana 也推出了功能完备的 iPhone 集成开发功能，在 Adobe 公司的 RIA 产品 AIR 推出不久之后，Aptana 就支持了 AIR 的开发环境。所以 Aptana 是 Web 开发工程师手中的一把利器，但是它仍然继承了 Eclipse 家族打开速度较慢、运行需要资源量较多等缺点。

如图 2-6 是 Aptana 的集成开发界面，整个界面默认背景色为黑色，代码默认色就是根据色彩明暗度变化比例，组成最利于编程人员的颜色。这种颜色搭配要比白色配黑色更能使眼睛不易疲劳。

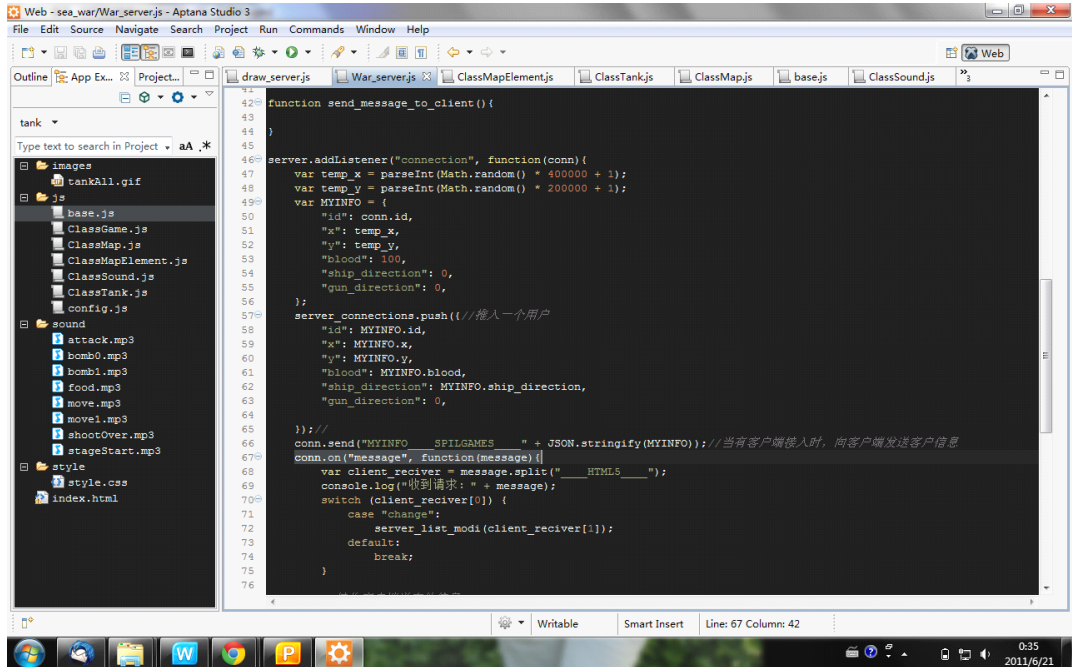


图 2-6 Aptana 集成编译环境界面图

2.2.2 浏览器发展和 Chrome 浏览器介绍

浏览器随着互联网的发展也在渐渐的变化着，如今的浏览器在性能和功能上都已经不能和曾经只是展示一张网页、只是解析 HTML 的时代相比，如今各大厂商都在争夺这个真正意义上的“互联网第一入口”。目前市面上的浏览器虽然品牌众多，但是其都是根据几大开源内核来构建，主要分为两大类：IE 系列和非 IE 系列^[16]。

IE 基于 Trident 排版引擎构建，由于微软的商业目的等诸多原因，IE 浏览器总是不去很好的实现 Web 的标准，导致很多 Web 工程师要针对 IE 进行很多单独的开发，而有些却只对 IE 浏览器开发，导致其他非 IE 浏览器的用户无法使用，这些问题导致了很大的资源浪费，不过这些即将在 HTML5 时代解决，以为微软也清楚，这是一个互联网的时代了。国内的很多厂商基于 IE 开发了一些类似扩展的浏览器，实际上就是在 IE 浏览器的外面加了一层壳，不过由于考虑到了很多用户的需求，所以很多功能受到用户的欢迎。国内基于 IE 做的较好的有遨游、360 浏览器、搜狗浏览器等^[17]。非 IE 系列的浏览器大都能够对标准

很好的实现，所以更注重的是性能提升和满足用户需求，其中以 WebKit 内核做的做好。

还有一个较为强大且影响深远的浏览器，那就是鼎鼎大名的——Firefox（中文名称火狐），是一个开源网页浏览器，使用 Gecko 引擎(即非 ie 内核)，由 Mozilla 基金会与数百个志愿者所开发。由于微软的 IE 占有统治地位，有垄断的倾向，所以需要有一个开源的而且又强大的浏览器来与之对抗，而这个时候恰恰 Firefox 诞生了，它基于开源的信仰，很快就发展起来了。基金会的支持，数百万开发者的响应，linux 操作系统爱好者们几乎都换用 firefox。强大而又可扩展，但是随着插件安装数量的增多，firefox 打开是速度会下降，打开速度慢一直是 firefox 遭受诟病的源头。

如图 2-7 为 firefox 浏览器架构图，从架构中即可看出，firefox 的架构过于庞大，但是也同样很强大，这样强大的架构使其显得有些臃肿，导致打开速度慢，用户体验不佳。

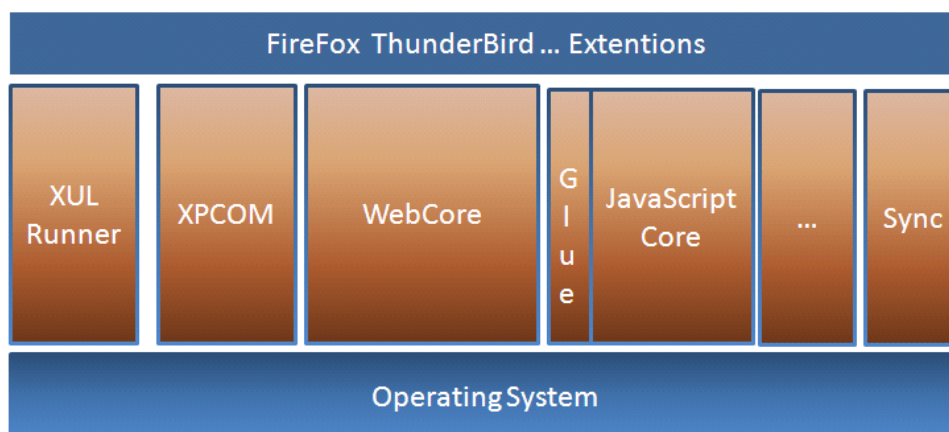


图 2-7 Firefox 浏览器架构图

基于 WebKit 构建的浏览器主要有 Google 的 Chrome、Apple 的 Safari 等。Webkit 已解析速度快著称，而且对标准鼎力支持，得到了绝大部分 Web 工程师的认可。而且 Webkit 中内建了一个开发者工具，工程师方便进行开发和调试。值得一提的是，国内的很多浏览器厂商看到了 IE 的问题，也在尝试基于双内核构建浏览器，如遨游 3、搜狗浏览器等。

Google 的 Chrome 浏览器基于 Webkit 构建，并且 Google 的工程师们开发了能够快速解析 Javascript 的 V8 引擎与 Webkit 搭档，使得 Chrome 浏览器打开

网页的速度极快，渲染性能也极强，而且能够很好的支持 HTML5。有 Google 的强大支持，基于开源项目构建的 Chrome 得到了越来越多的用户和开发者的认可。基于 Chrome 的 Chrome 软件商店里面有大量应用，使得 Chrome 更加的强大。但是由于国内的特殊环境及技术原因，国内开发的很多应用和站点不支持非 IE 内核的浏览器。如图 2-8 所示为 Chrome 浏览器架构图，可以看出 Chrome 的架构很小巧，而且每个模块又是性能强劲，合理的组织到一起，使 Chrome 整体性能表现出色，打开速度快，用户体验较佳。

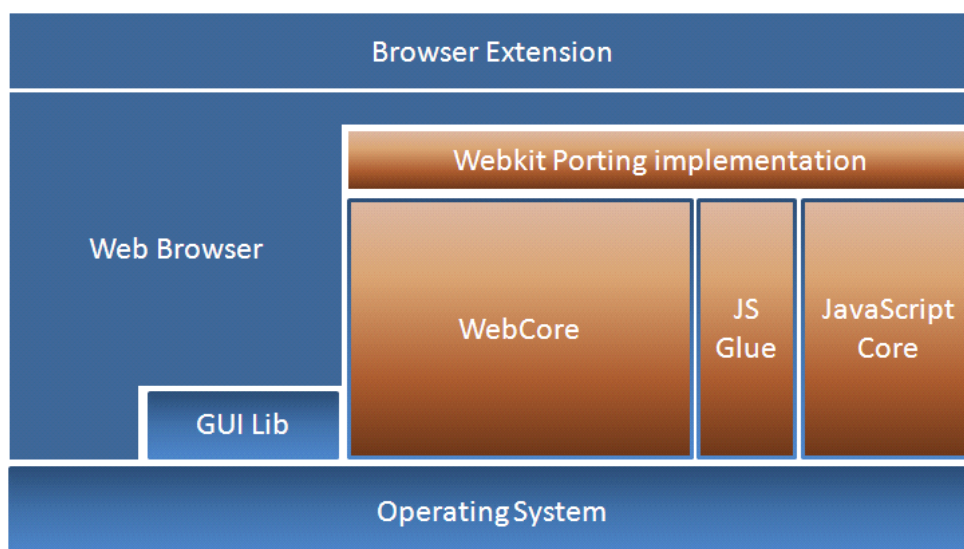


图 2-8 Chrome 浏览器架构图

第3章 游戏玩法设计和需求分析

3.1 游戏玩法设计

根据网络实时性特点和技术复杂度，设计游戏玩法如下：

用户可以通过浏览器打开游戏，游戏打开后，用户被自动分配到红方或者绿方阵营中，目的就是消灭对方阵营的坦克。游戏中被敌方坦克击中，就算死亡，相应对应的成绩会加一。

3.2 需求分析

3.2.1 系统角色

游戏玩家：玩游戏的人，能够操作坦克在地图上面行动和发射炮弹。

3.2.1 用例分析

玩家及玩家所具有的功能，如图 3-1。

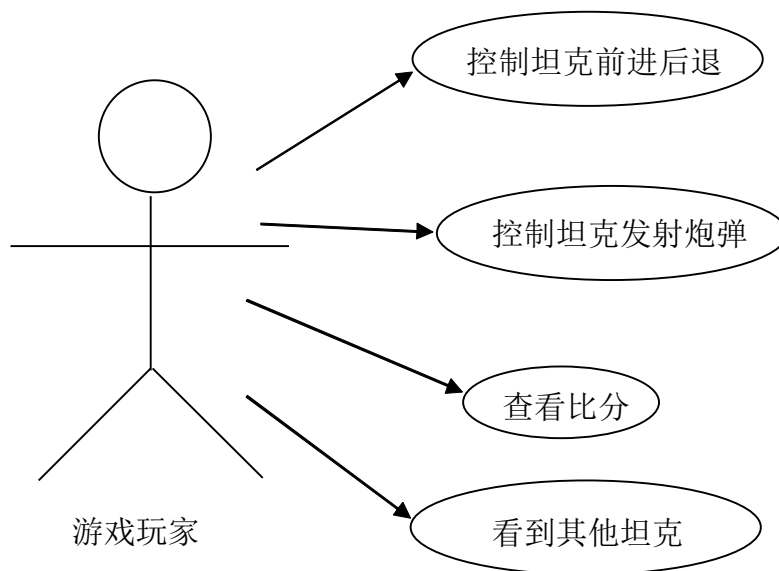


图 3-1 用例图

第 4 章 总体设计及程序功能分析

4.1 程序总体功能设计

4.1.1 程序功能模块分析

该游戏主要分为两大部分：

- (1) 服务器端：主要负责数据的集中管理和数据的分发。
- (2) 客户端：主要负责用户操作，游戏界面绘制、实现游戏玩法、数据的发送和接收等功能。

如图 4-1 为坦克大战游戏的总体模块图。

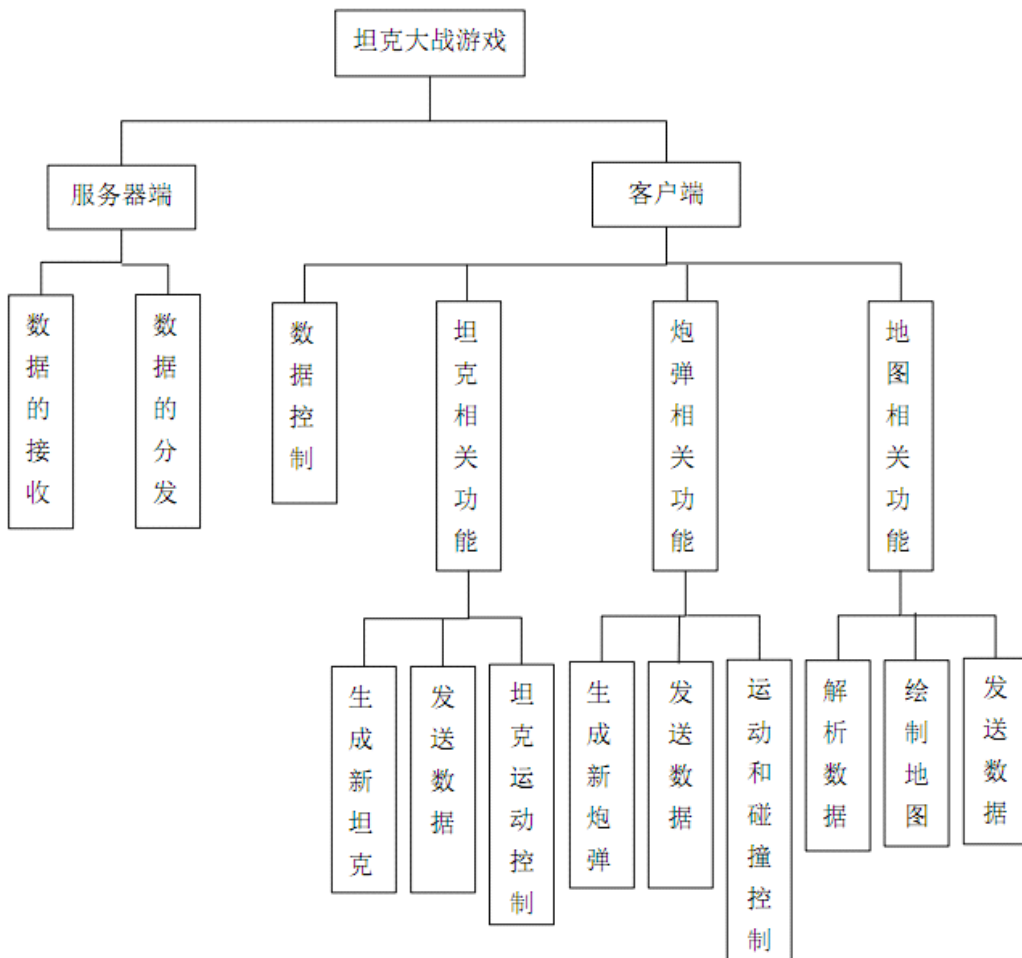


图 4-1 坦克大战游戏总体模块图

4.1.2 地图读取及共享分析

游戏要能够较容易地自由的修改界面上形成的游戏地图，且还可以灵活地改变整个游戏的大小，实现联机对战的所有玩家在地图生成和改变时都能够实时的共享整个游戏地图。

4.1.3 声音的控制分析

根据游戏设计播放声音，并且部分声音玩家按键控制坦克时播放，游戏可以自如的播放出声音，且较为流畅。

4.1.4 联网对战分析

玩家可以通过浏览器直接打开游戏，并且能够和其他玩家联机对战，实时看到所有玩家的坦克。要想实现这种功能，我们就要先分析一下现有的技术和架构如何来实现。目前，处理客户端之间的交互有两种主要架构。

(1) 对等网络架构 (Peer-to-Peer, P2P)。信息只在客户端间传送，不需要服务器的介入。可能首先会使用服务器让客户端间能查找到对方，但在此之后就不再需要服务器了。具体的可以分为两种不同的形式：

①完整连接拓扑架构。每个客户端与其他每个客户端之间都必须有连接，信息可以直接在用户间交换。

②环状拓扑架构。信息只有流经一个或多个客户端后才能传递过来。

P2P 架构的延时较小，在 C/S 模式中，信息首先是从一个客户端传到服务器端，然后再从服务器端传到另一个客户端。但 P2P 架构则是让信息直接在两个客户端间传递，这样就比 C/S 模式减少了一半的传递时间。另外，P2P 不需要服务器，这对于无论是游戏开发者还是发布者来说都是一件好事，因为不用再为维持运行游戏中央服务器而支付主机托管和管理费用了。

但是同时 P2P 也有很多的缺点。首先当游戏的客户端相当少的时候，P2P 架构的延时很小且不需要使用服务器端。但是当游戏有很多客户端时，P2P 架构就不适用了，因为他需要每个客户端都和其他客户端保持一个独特的链接，建立了很多链接，性能较差，这些都导致 P2P 游戏的可扩展性差。

(2) 客户端-服务器架构 (Client-Server, C/S)。客户端只把信息传送到服务器上，然后服务器再把信息传给相应的客户端。

Socket 服务器可于某处运行并监听连接尝试，接受并管理来自客户端的连

接，智能规划客户端间的信息传送路线。Socket 服务器依托一个 IP 地址或主机名称，在至少一个端口上监听连接尝试。客户端一旦与 Socket 服务器建立起连接，也就和服务器建立起了持续不断的 Socket 连接。通过这个开放的 Socket 接口，客户端能够将信息传递给服务器，反之亦然。这种 Socket 是开发并且随时可用的，所以信息互相传递非常快捷，更新是基于事件驱动的而且是实时的。

所以，本课题采用 C/S 架构的 Socket 服务器方式来实现服务器端程序。

4.2 程序流程

根据功能模块划分，程序主要分为三部分各自独立的流程：

(1) 服务器端流程。服务器端流程每次都由接收到客户端数据改变作为事件触发的句柄，之后发送给所有客户端。

服务器端流程如图 4-2 所示。

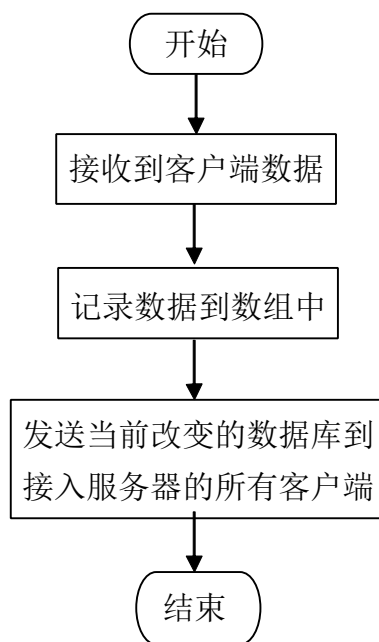


图 4-2 服务器端流程图

(2) 客户端界面控制流程。所有程序由定时器触发，每次触发就会去重新绘制当前 Canvas。第一次创建时，首先要连接到服务器获取到地图信息，之后再绘制地图，生成坦克。

如图 4-3 所示，为客户端打开后连接服务器生成地图流程图。

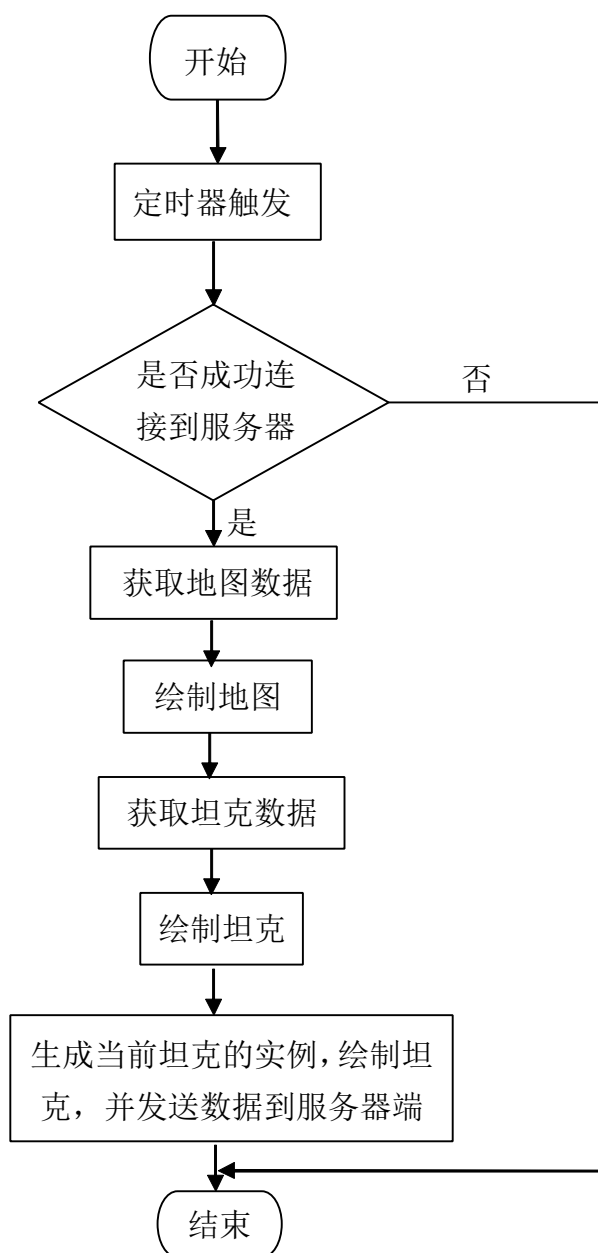


图 4-3 客户端打开后连接服务器生成地图流程图

(3) 客户端数据发送流程。每次当坦克位置发生变化，或者地图发生变化，则会触发向服务器发送数据的方法。

如图 4-4 所示，为客户端数据发送流程。

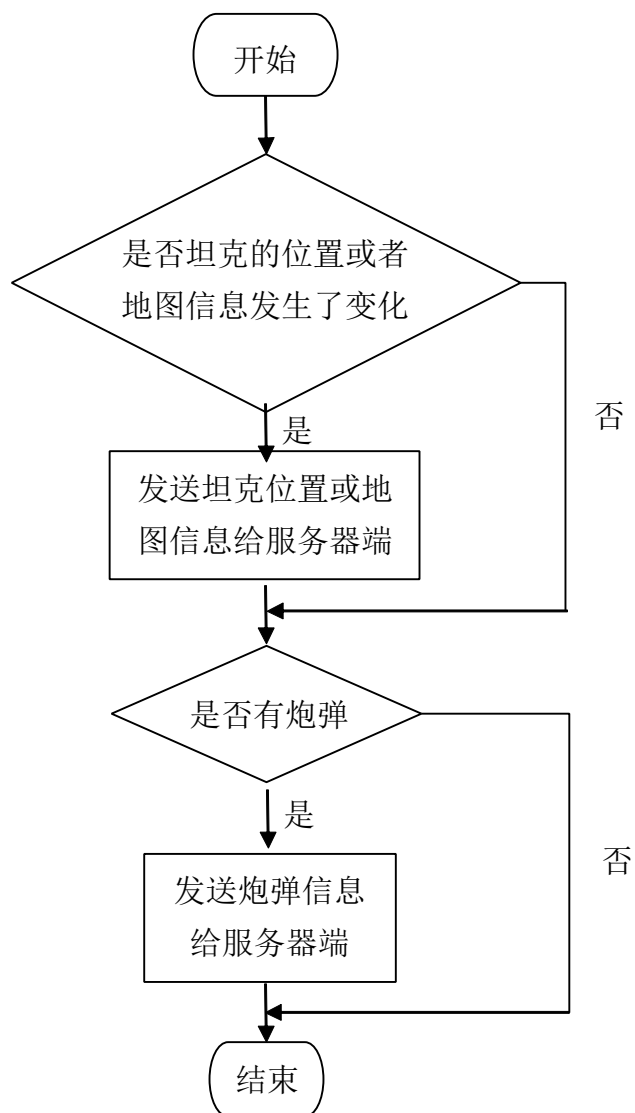


图 4-4 客户端数据发送流程

4.3 模块及对应的方法

主要模块分为服务器端和客户端：

(1) 服务器端模块：

服务器端模块具体功能有调用服务器 Nodejs 模块、调用 WebSocket Server 来提供服务器端基础应用模块。然后通过具体方法处理客户端的连接、数据的接受和分发。

如图 4-4 为服务器端模块所包含的具体方法。

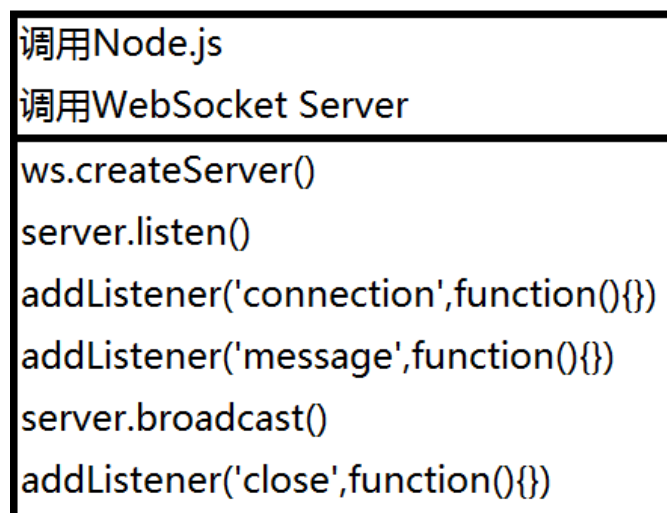


图 4-5 服务器端模块方法图

(2) 客户端模块:

客户端主要负责用户操作，游戏界面绘制、实现游戏玩法、数据的发送和接收等功能。分为基础类模块 ClassBase（如图 4-5）、游戏主函数模块 ClassGame（如图 4-6）、地图模块 ClassMap（如图 4-7）、游戏坦克模块 ClassTank（如图 4-8）和声音控制模块 ClassSound（如图 4-9）。

①base 模块中有 loadImage 方法，该方法可以读取图片，并且在图片加载完成后，异步调用传入的其他方法。

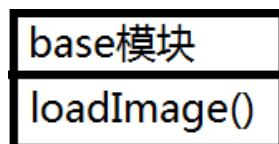


图 4-6 客户端 base 模块方法图

②ClassGame 模块是整个游戏运行的主逻辑住调用的类，其中有主入口函

数 `init()`、当服务器器打开时触发方法 `onopen()`、接收数据方法 `onmessage()`、键盘事件控制方法 `control()`、游戏开始 `start()`和游戏总的时间触发器 `move()`。

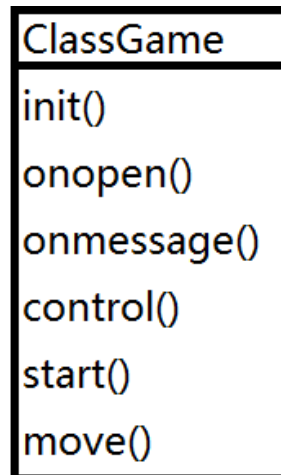


图 4-7 客户端游戏总控制 Game 模块图

③ClassMap 模块是地图相关方法模块，其中主要有清除某一位置地图方法 `clean()`、清除所有地图方法 `cleanAll()`、绘制某一位置地图方法 `draw()`、绘制全部位置地图方法 `drawAll()`、取得某位置 z-index 数值方法 `getZ_index()`等。

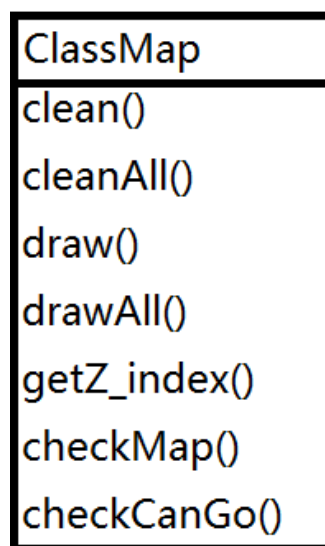


图 4-8 客户端地图相关 Map 模块图

④ClassTank 是玩家控制的 Tank 的主要方法的类,其中有生成新坦克的 birth 方法、取得下一步的方法 getNextStep()和绘制坦克 drawTank()等方法。

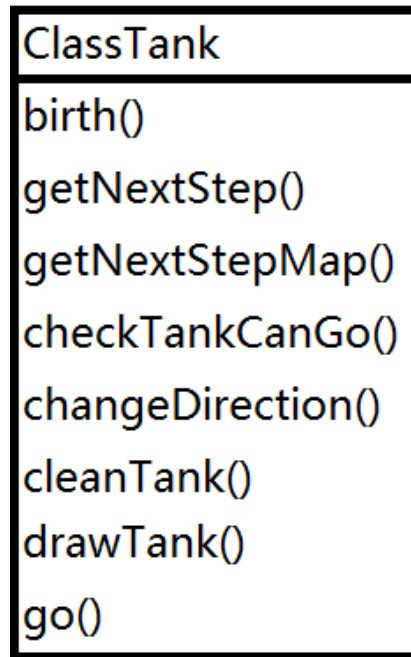


图 4-9 客户端 tank 模块图

⑤ClassSound 方法是声音控制类,其中有基本的声音控制方法,播放声音 play()和暂停播放 pause()等方法。

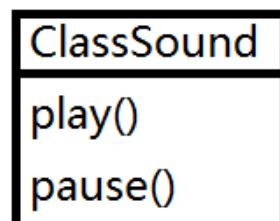


图 4-10 客户端声音相关 Sound 模块图

第 5 章 详细设计

5.1 服务器端设计

5.1.1 服务器端的运行环境

服务器端环境使用的是 Cygwin 在 window7 操作系统下内建一个 linux，类似于在虚拟机中安装 linux 环境。基于此安装 Nodejs 的包管理 NPM，之后安装并配置 Nodejs。除了 Nodejs 运行环境外，还需要一个 Web 发布环境，本课题采用 Apache 作为 Web 页面发布程序^[18]。

5.1.2 服务器端的设计

(1) 服务器端需要保证实时性和统一性，所以首先我们要考虑下需要同步的数据量：

①地图数据。地图数据需要先写好在服务器端，当用户连接时才能保证每台机器的数据都是同一的，每台客户端都先请求地图数据，然后绘制地图。

②坦克数据。每个客户端都要能够实时的获取到其他客户端的数据信息，这样才能够实现实时互联。

③炮弹信息。每个客户端都要能够发射出炮弹，而炮弹信息也需要实时传送。

所有这些信息的改变都是基于事件触发的，这样才能够有效的减少服务器与客户端之间传输的数据量，并且又不影响较高的实时性^[19]。

(2) 实现方法：服务器端主要使用事件触发机制的几个关键方法来完成数据共享，方法如下：

①当有新客户端连接之后触发的事件：

```
server.addListener("connection", function(conn){}).
```

②接收到客户端发来的数据时触发的事件：

```
conn.addListener("message", function(message){}).
```

③向所有客户端分发数据的事件：

```
server.broadcast(message).
```

5.1.3 服务器端部分代码

以下代码实现了服务器端的基本功能，和程序的基本数据传输功能：

```
var http = require('http');
var sys = require('sys');
var ws = require('/home/test/node-websocket-server/lib/ws/server');
var os=require('os');
var fso = require('fs');
var port=8000;//端口号
var speed=30;//游戏速度
var server = ws.createServer();//建立 websocket
server.listen(port);//监听端口
console.log('服务器端程序开始运行，端口号是：'+port+'，请注意设置好客户端访问 IP 和端口号！');//输出服务器运行消息
var tankList=[];//储存坦克信息
var flag=0;
//地图初始数组
// 0 是空，1 是红砖，2 是水泥墙，3 是草，4 是河
var mapArray=[
    [0,0,1,1,2,2,3,2,3,0,1,2,3,0,1,2,2,3,2,3,0,1,2,3,0,1,2,2,3,2,3,0,1,2,3,0,0,0,0],
    [0,0,1,1,2,2,3,2,3,0,1,2,3,0,1,2,2,3,2,3,0,2,3,0,1,2,2,3,2,3,0,1,2,3,0,0,0,0]
];
server.addListener("connection", function(conn){
    tankList.push({id:conn.id,dx:0,dy:0});
    var widthNum=mapArray[0].length;
    var heightNum=mapArray.length;
    conn.send(widthNum+"."+heightNum+"."+mapArray);
    console.log(conn.id);
    conn.addListener("message", function(message){
        server.broadcast(message);//向全体发送信息
    });
});
```



```
server.addListener("close", function(conn){
    server.broadcast('Quit');
});
//console.log("");
```

5.2 地图功能模块

5.2.1 读取地图

地图数据存储在服务器端的一个数组变量中，地图是一个二维平面图，所以采用二维数据来存取^[20]。客户端从服务器端获得地图数据之后，在客户端解析、写入到客户端的一个数组中，然后从中读取地图信息，并且根据数组对应的数据，绘制出地图。

绘制出的地图效果如图 5-1。

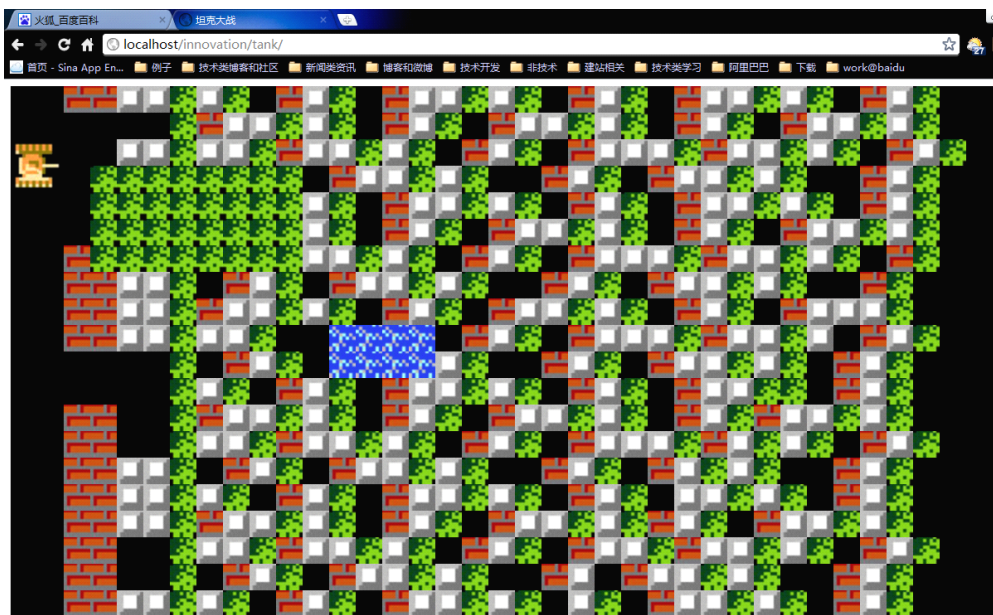


图 5-1 坦克大战绘制地图后的界面图

5.2.2 绘制地图

每个客户端都有识别地图并绘制的方法，绘制地图的前提是解析地图信息。地图信息变量是一个二维数组，数组每个位置上对应一个数字，即为当前地图

元素的地图信息^[21]。

本课题中地图数据及对应的信息：

0 代表空（null），1 代表红砖，2 代表水泥墙，3 代表草，4 代表河。

每一个地图元素都随之对应了相应的地图属性：

- (1) 空（null）。物体全部能够通过，元素为黑色填充矩形。
- (2) 红砖。物体不能穿过，炮弹可打击，打击后消失。
- (3) 水泥墙。物体不能穿过，炮弹不可打击。
- (4) 草。物体全部能够穿过，但是绘制图像要在草元素的下面。
- (5) 河。坦克不可穿过，炮弹可穿过但不可打击。

5.2.2 地图模块部分代码

以下代码实现了地图基本的清除和绘制地图的功能：

```
Map.prototype={
  //清除指定地点的地图
  clean:function(dx,dy){
    j=Math.floor(dx/this.dw);
    i=Math.floor(dy/this.dw);
    this.mapElementArray[i][j].clean();
    //this.map_array[i][j]=0;
  },
  //清除全部地图
  cleanAll:function(){
    context.fillStyle='#000000';
    context.fillRect(0,0,this.width,this.height);
  },
  //绘制指定地点的地图
  draw:function(dx,dy){
    j=Math.floor(dx/this.dw);
    i=Math.floor(dy/this.dw);
    this.mapElementArray[i][j].draw();
  },
}
```

```
//检测指定位置是什么地形
checkMap:function(dx,dy){
    j=Math.floor(dx/this.dw);
    i=Math.floor(dy/this.dw);
    return this.map_array[i][j];
}
```

5.3 坦克模块的设计

5.3.1 读取服务器端信息

服务器端有其他坦克的数据信息传入，则会触发读取坦克数据信息的方法，方法会将客户端中记录所有坦克信息的对象中的值改变^[22]。当当前客户端下一次重新绘制界面的时候，对应的坦克就会按照新的数据来绘制。

5.3.2 控制坦克

坦克通过主游戏函数中的 `control` 方法，`control` 方法会调用键盘监控按键事件，如果用户按下对应的键，触发的事件会去修改变量 `direction` 中的方向值，当下次客户端定时器触发了重新绘制的事件时，首先调用 `Tank` 类中的方法读取 `direction` 并且修改对应的量，然后绘制坦克，这样坦克就改变的位置，实现用户控制坦克的效果。而每次改变时会触发向服务器端发送当前坦克信息的事件，服务器接受到之后就会分发给其他所有客户端^[23]。

5.3.3 绘制画面

根据地图元素的信息，绘制对应地图图像。首先要判断地图元素中的，`z-index` 属性，如果地图该属性大于当前位置的其他元素，则先绘制其他元素。其他元素的绘制方式也如地图。该方法有定时器触发^[24]。

5.3.4 坦克模块的部分实现代码

以下代码实现了坦克的基本运动功能：

```
Tank.prototype={
    //生成一个新坦克
```

```
birth:function(dx,dy,map){
    var startSound=new Sound('start');
    startSound.play();
    this.dx=dx;
    this.dy=dy;
    this.map=map;
    this.maxWidth=map.width;
    this.maxHeight=map.height;
    this.drawTank(this.dx,this.dy);
},
//改变坦克的朝向
changeDirection:function(direction){
    switch(direction){
        case 'up':
            this.sx=this.up;
            break;
        case 'down':
            this.sx=this.down;
            break;
        case 'left':
            this.sx=this.left;
            break;
        case 'right':
            this.sx=this.right;
            break;
    }
    this.drawTank(this.dx,this.dy);
}
```

5.4 炮弹模块的功能

5.4.1 读取服务器端信息

服务器端有其他炮弹的数据信息传入，则会触发读取炮弹数据信息的方法，方法会将客户端中记录所有炮弹信息的对象中的值改变。当当前客户端下一次重新绘制界面的时候，对应的坦克就会按照新的数据来绘制。

5.4.2 碰撞检测

当炮弹元素内部坐标改变时，都会去判断下一步中坦克的位置和地图信息，如果地图非空元素或者坦克元素出现，则会调取对应的处理方法来完成碰撞检测过程。该事件也有定时器触发。

5.4.3 击毁功能

当判断出下一步是地图非空元素或者坦克元素时，则会消灭该元素。如果是地图元素，则会改变地图数组中对应的值，并发送到服务器端，服务器则会分发给所有客户端。这样在下次重绘界面时，地图就被改变，即被“击毁”。坦克也是如此，读取坦克数据中的信息，然后修改。

5.4.4 绘制画面

全部游戏界面的绘制由定时器定时触发，每次触发就会调用对应模块中修改数据的方法，然后绘制对应元素。

5.5 客户端与服务器端通信

客户端与服务器端通信全部通过 HTML5 中的 WebSocket 方式，当用户触发了对应发送数据的方法时，客户端程序就会和服务器端通信，发送当前改变的数据。

主要的 WebSocket 通信方法：

(1) 当客户端与服务器端连接时触发事件

`s.onopen = function(event){}`。

(2) 客户端接受到服务器端发送的数据

`s.onmessage = function(event){}`。

(3) 客户端关闭

`s.onclose = function(event) {}`。

(4) 客户端发送数据到服务器端

`s.send()`。

第 6 章 结论

通过本课题对基于 HTML5 技术的研究和实践，完成了一个纯粹由 WEB 技术中开放的 HTML5 技术标准和 Javascript 语言实现的实时性对战游戏平台。这充分的说明了下一代互联网标准 HTML5 的强大能力，并且直接通过浏览器打开的应用具有非常好的用户体验和较高的客户端安全性，这是其他的技术所不具备的优势。

在课题的研究和实践中遇到了资料不足，技术 API 总是修改等诸多的问题，但是也正是通过互联网，秉承着互联网中开放、分享、共同成长的精神，和网上有着同样爱好的技术人员交流和学习，困难都被成功的解决，可见互联网的意义和价值。而用来构建下一代互联网的技术又是这么的强大，所以未来的互联网将会发展的更为强大，能够带给人们更多意想不到的体验。

目前各大浏览器厂商也加大研发力度，全部发力于 HTML5。看到了 Iphone 和 Ipad 上的 Safrai 浏览器、看到了 Opera mobile 浏览器，我看到了 HTML5 未来在移动客户端的前景和发展，未来的互联网不仅仅是属于 PC 的，而是属于所有用户的。这些用户可能用平板电脑、可能是 Android 手机的用户、可能是 Iphone 的用户或者是某种现在还没有的设备的用户，但是可以肯定他们都将会更好的接入互联网，更方便快捷的访问互联网中的信息，加之 IPv6 的全面使用，互联网基础架构层以实现了技术的改造和迁移。而客户端的 HTML5 正是用来解决应用层技术架构的问题，技术的发展推动着模式的创新，而不断的创新又不断的孕育出新的技术，这样的良性的循环在互联网这个前沿、高科技并且产品迭代速度快的行业中尤其突出。

参 考 文 献

- [1] 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.
- [2] 皮尔格林, 常可, 胡金埔, 赵静. HTML5 揭秘[M]. 北京: 电子工业出版社, 2010.
- [3] 罗森, 夏普. HTML5 用户指南[M]. 北京: 机械工业出版社, 2011.
- [4] 刘红伟. HTML5 和样式应用指南[M]. 北京: 机械工业出版社, 2011.
- [5] 陆凌牛. HTML5 与 CSS3 权威指南[M]. 北京: 机械工业出版社, 2011.
- [6] 泽卡斯. JavaScript 高级程序设计(第 2 版)[M]. 北京: 人民邮电出版社, 2010.
- [7] 费拉纳提. JavaScript 权威指南(第五版)[M]. 北京: 机械工业出版社, 2007.
- [8] 月影. JavaScript 王者归来[M]. 北京: 清华大学出版社, 2008.
- [9] 亚当斯. JavaScript 艺术与科学[M]. 北京: 电子工业出版社, 2010.
- [10] 泽卡斯. 高性能 JavaScript[M]. 北京: 电子工业出版社, 2010.
- [11] 克罗克福德. JavaScript 语言精粹[M]. 北京: 电子工业出版社, 2009.
- [12] 哈梅斯, 迪亚斯. JavaScript 设计模式[M]. 北京: 人民邮电出版社, 2009.
- [13] Jeanine Meyer. The Essential Guide to HTML5: Using Games to learn HTML5 and JavaScript.[M]. 2010.
- [14] Julie C. Meloni, Michael Morrison. Sams Teach Yourself HTML and CSS in 24 Hours[M]. 2010.
- [15] Matthew David. HTML5: Designing Rich Internet Applications[M]. 2011.
- [16] Brian P. Hogan. HTML5 and CSS3: Develop with Tomorrow's Standards Today[M]. 2010.
- [17] Peter Lubbers, Brian Albers, Frank Salim. Pro HTML 5 Programming: Powerful APIs for Richer Internet Application Development[M]. 2010.
- [18] Mark Pilgrim. HTML5: Up and Running[M]. 2010.
- [19] Rob Hawkes. Foundation HTML5 Canvas: For Games and Entertainment[M]. 2010.
- [20] Steve Fulton, Jeff Fulton. HTML5 Canvas[M]. 2011.
- [21] Bill Sanders. Smashing HTML5[M]. 2011.

-
- [22] Christopher Murphy, Richard Clark, Oli Studholme. Beginning HTML5 and CSS3: The Web Evolved[M]. 2011.
- [23] <http://baike.baidu.com/view/951383.htm>[EB].
- [24] Ding Kong, Gloria Ng, Jimmy Lo, Johan Harjono. Building smarter web applications with HTML5[A].Proceedings of the 2010 Conference of the Center for Advanced Studies on Collaborative Research (CASCON '10)[C], 2011. 402-403 .

致 谢

衷心感谢各位!