**学科分类号** 0809



**本科生毕业设计**

**题目（中文）:** 基于Canvas的web版打砖块

游戏的设计与实现

**（英文）:** Design and Implementation of

Arkanoid game based on

Canvas Framework

**学生姓名：** 叶志文

**学 号：** 1500130604

**院 别：** 计算机科学与工程学院

**专 业：** 软件工程

**指导教师：** 邓邵伟 讲师

**起止日期：** 2018.06-2019.05

**2019 年 5 月 10 日**

**怀化学院本科毕业设计诚信声明**

作者郑重声明：所呈交的本科毕业设计，是在指导老师的指导下，独立进行研究所取得的成果，成果不存在知识产权争议。除文中已经注明引用的内容外，设计不含任何其他个人或集体已经发表或撰写过的成果。对设计的研究做出重要贡献的个人和集体均已在文中以明确的方式标明。本声明的法律结果由作者承担。

本科毕业论文设计作者签名：

年 月 日

目 录

[摘　要 I](#_Toc8598690)

[关键词 I](#_Toc8598691)

[Abstract I](#_Toc8598692)

[Key words I](#_Toc8598693)

[1 前言 1](#_Toc8598694)

[1.1研究背景与意义 1](#_Toc8598695)

[1.1.1 研究背景 1](#_Toc8598696)

[1.1.2 研究意义 1](#_Toc8598697)

[1.2 研究现状 1](#_Toc8598698)

[1.2.1 国内研究现状 1](#_Toc8598699)

[2 关键技术概述 1](#_Toc8598700)

[2.1 HTML5简介 1](#_Toc8598701)

[2.2 Canvas框架概述 2](#_Toc8598702)

[2.3 开发环境 2](#_Toc8598703)

[3 需求分析 2](#_Toc8598704)

[3.1 系统功能需求 2](#_Toc8598705)

[3.2 系统非功能需求 3](#_Toc8598706)

[4 系统的设计 3](#_Toc8598707)

[4.1 系统的体系结构 3](#_Toc8598708)

[4.1.1 游戏界面绘制模块 3](#_Toc8598709)

[4.1.2 游戏操作模块 4](#_Toc8598710)

[4.1.3 游戏调试模块 5](#_Toc8598711)

[5 系统的实现 6](#_Toc8598712)

[5.1 游戏框架的设计 6](#_Toc8598713)

[5.2 scene场景的实现 8](#_Toc8598714)

[5.3主场景控件的实现 11](#_Toc8598715)

[5.4 按键监听模块实现 13](#_Toc8598716)

[6.1 测试计划 15](#_Toc8598717)

[6.1.1 测试进度 16](#_Toc8598718)

[6.1.2 测试方案 16](#_Toc8598719)

[6.1.3 测试目的和原则 16](#_Toc8598720)

[6.1.4 测试环境 16](#_Toc8598721)

[6.2 测试说明 16](#_Toc8598722)

[6.2.1 游戏界面显示测试 16](#_Toc8598723)

[6.2.2 挡板移动控制 18](#_Toc8598724)

[6.2.3 小球移动控制 19](#_Toc8598725)

[7 测试及性能分析 19](#_Toc8598726)

[7.1 测试结果概述 19](#_Toc8598727)

[7.2 测试结果评价 20](#_Toc8598728)

[8 总结与展望 20](#_Toc8598729)

[8.1 本论文研究成果 20](#_Toc8598730)

[参考文献 20](#_Toc8598731)

[致 谢 21](#_Toc8598732)

基于Canvas的web版打砖块游戏的设计与实现

摘　要

随着科技的进步，人们更希望通过网络、通过游戏来释放自己的压力，以往的技术都只是让游戏在笔记本或者移动产品上单独运行，并不可以跨平台操作，这也增加了开发者的工作量，因此经过人们不懈的努力，终于在原有基础上对html进行了改革，这就是HTML5。 不仅如此HTML5中的Canvas元素可以使浏览器直接创建并处理图像，减轻了开发人员的负担，减少了用户刷新页面的时间，得到了更多人的认可,因此基于HTML5所制作的游戏也就成了很多人的研究课题。  本文通过介绍HTML5的一些特性，并通过当下比较流行的打砖块游戏的研发过程分析来使HTML5的内容更加形象，通过设计并开发打砖块这个游戏，让我对JSP这门脚本语言以及H5的Canvas标签有了更深的理解。

关键词

游戏；HTML5；Canvas；JSP；

Design and Implementation of Arkanoid game Based on Canvas Framework

Abstract

With the progress of science and technology, people want to release your stress through the network, through the game, in the past, technology would just let the game run independently on a laptop or mobile products cannot cross - platform operation, which has increased the workload of the developers, and through the concerted efforts of the people, finally reform on the existing basis for HTML, this is html5.Not only that canvas elements in html5 can be created directly by the browser and image processing, reduce the burden on developers, and make the interface more beautiful, with a very good user experience.Reduces the time the user refreshes the page, endorsed by more people, so the game is based on the html5 produced a lot of research topics.This article by introducing html5 features, and by now more popular arkandroid analysis of the development process to make the html5 more images, design and development by hitting the bricks of the game, let me this portal for JSP scripting language, as well as a deeper understanding of the H5 canvas tag.

Key words

Game；HTML5；Canvas；JSP；

1 前言

1.1研究背景与意义

1.1.1 研究背景

相比传统端游，HTML5应用可以绕开应用市场的限制进行自主实时更新，用户可以快速享受新服务。而且这种更新完全可以是差量更新，比如某个HTML页面或某个js文件有问题，只更新这个几k的小文件就可以了，这比原生应用的更新体验好太多。H5游戏不同于传统意义的端游，不需要下载客户端，随开随玩，这也是H5最大的优点，便捷。H5游戏可以通俗地成为网页游戏，只是与传统网页游戏不同的是，H5游戏具有更好的特性以及更强的技术支持。特别是当手机平台浏览器的硬件加速功能普及的今天，借助手机的流行，H5游戏正在走红，越来越多的好游戏出现。和传统手游不同，H5手机游戏的一大优势在于无需客户端加持，即点即玩，对用户网络质量的要求更低。H5游戏，以其不用下载、免安装、不占内存的优势，受到越来越多游戏玩家的青睐。所以我想开发一个基于Canvas的打砖块游戏加深对网页游戏原理的理解。

1.1.2 研究意义

此项目的目的在于通过完成游戏项目，加深对JSP的语法、算法等多方面的学习，深入了解图形用户界面的设计，并且熟练的使用编程软件，将书本上的理论知识应用到实践当中，来拿这款打砖块小游戏来验证下自己的学习的理论知识是否扎实。希望通过这一次的毕业设计，让我在程序的设计和算法的设计上有进一步的认识，然后获得一定提高。通过设计增加了我们的实际动手能力，对于我们以后工作能力和信心的培养有重要意义，进而与社会所需的人才相接轨。更好的完善自己！

游戏是一种艺术，而JSP是一种技术，集合不同的艺术门类，并把艺术与技术相结合，完成游戏的开发，游戏可以让人在疲倦或休闲之时获得轻松而又愉快的享受。所以，开发出大家都比较喜欢的，高品质的，简单容易好玩的休闲游戏，将会受到人们的普遍欢迎。我认为H5作为一项兴欣技术，网页，简单方便，做游戏有其不能替代的优势。

1.2 研究现状

1.2.1 国内研究现状

JSP编程语言具有非常多的优点，它简单易用，不用配置环境，不需要编译，可直接在浏览器上运行。

游戏的产业是新时代发展起来的行业，它的独特的力量使它在当今的娱乐领域占有一块非常大的比重，当前的种种原因使得用户可以选择的娱乐活动变少，这是游戏产业发展起来的一个原因，但是更大的原因是游戏具有非常大的吸引力。如今，小游戏、网页游戏深受广大网民欢迎，它方法简单，不需要繁复的注册下载等过程。 HTML5对旧的Web网页制作技术做了大量的改进、创新。它增加了很多新的元素，将一些模块动态化， 国内QQ空间应该是市面上最早布局H5游戏的平台，当年2014年初，摸索了一段时间后，原本负责H5平台整体运营的同学改变了方向转做手游分发，H5一直的产业困境，最直接表面的原因是，没有稳定持续的，足够多的，好的 H5游戏产品面世。H5游戏经过几年的发展，持续稳定输出的游戏产品是有了，但还不够大量做H5游戏的要吧是名不见经传的小CP，做出来的游戏产品质量也就那样，并没有特别脍炙人口的作品。当然，H5作为一门网络语言，也有其天然的几个问题，游戏的体验本身，包括画面，进程的流畅程度等，H5对比原生确实有差距，玩家游戏过程中的体验决定了后续的转化。但这其实是最小的问题了，技术上来后就可以。所以目前国内H5游戏还是一个迅猛发展的状态。

2 关键技术概述

2.1 HTML5简介

HTML5是万维网html语言的第五个版本，目前比较流行的版本是HTML4.01，虽然相较先前的版本有了很大的改善，但是随着移动应用的发展，以前的那些版本已经远远达不到人们对于科学技术的要求，Html需要在包括语言方面，语法方面和UI、API等各方面的更新，因此就产生了现在这个新的版本。  在HTML5之前人们更注重用html来制作一些静态的网页，那时候人们通过网络的交互还不是那么发达，而HTML5将音频，视频，动画等内容融为一体，更是增加了很多新的元素标签，去除了一些繁琐，冗长的标签，将一些相似标签融为一体，比如增加了表单验证标签，节省了表单输入时的麻烦，这些新的标签是网页结构更加紧凑合理。当然对于游戏来说最主要的还是图形动画的界面要美观，这样才可以吸引更多的人来玩，对于这方面不得不提的就是Canvas标签了，它与其他元素不同，不像audio等元素直接将现有元素插到网页中，而是可以独立的处理或创建2D图形，不仅如此，还可以通过JavaScript语言来控制Canvas的图形来响应与用户交互的动态图形与动画。

2.2 Canvas框架概述

<canvas> 是 HTML5 新增的元素，可用于通过使用JavaScript中的脚本来绘制图形。例如，它可以用于绘制图形，制作照片，创建动画，甚至可以进行实时视频处理或渲染。

Canvas 是 HTML5 标准中新晋的标签元素，它还提供了基于 JavaScript 的 2D 和 bitmap 图形的 API。

WebGL也使用<canvas>元素在网页上绘制硬件加速的3D图形。

2.3 开发环境

Atom是Github开发的一款免费的、功能非常强大的编辑器，它可以通过安装包的方式来扩展其功能。可以使用它来开发很多应用，例如：Web前端、JavaScript、CSS、HTML、Python、Ruby等等，Atom的官方网站是：https://atom.io/，进入网站，根据你的系统类型下载相应的安装文件。Atom的安装非常简单，基本上就是一路下一步，傻瓜式安装。

Git是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理。Git 是一个分布式版本控制系统，分布式相比于集中式的最大区别在于开发者可以提交到本地，每个开发者通过克隆（git clone），在本地机器上拷贝一个完整的 Git 仓库。而集中式版本控制系统最大的毛病就是必须联网才能工作，如果在局域网内还好，带宽够大，速度够快，可如果在互联网上，遇到网速慢的话，可能提交一个 10M 的文件就需要 5 分钟，这还不得把人给憋死啊。 当然，Git 的优势不单是不必联网这么简单，后面我们还会看到 Git 拥有极其强大的分支管理，把 SVN 等远远抛在了后面。

3 需求分析

通过调查互联网上已有的打砖块游戏、对游戏爱好者与开发者进行问卷调查，分析出网站所需要拥有的功能需求，确定了游戏的功能性需求与非功能性需求。

3.1 系统功能需求

通过分析调查，打砖块游戏系统界面的布局应尽可能的做到合理、美观。 具体针对本次设计，可以概括出功能大致如下：

（1）游戏基本功能（挡板的操作）：挡板左右移动、发射小球，通过键盘操作。

（2）游戏界面：开始画面，得分，难度，关卡，结束画面等。

（3）游戏随关卡等级提升使难度有所增加。

（4）游戏分数的统计。

（5）暂停游戏/继续游戏、退出游戏。

（6）砖块消除，挡板与小球碰撞，关卡编辑等功能。

（7）游戏能流畅的运行 ，通过小球碰撞砖块，消灭砖块同时能增加得分，能够手动跳掉下一关。通过最后一关后游戏胜利，如果挡板没有在小球下落时挡住小球，游戏结束。

3.2 系统非功能需求

 （1）可拓展性

产品在当前状态下，由于产品的业务需求可以增加或减少模块功能的能力，拓展性强的产品应对变化的能力更强。。

 （2）模块划分清晰

每个模块功能相互独立，方便用户的使用。

 （3）网站可靠性良好

网站处理的用户操作十分复杂，因此很可能出现异常的情况。产品的不稳定性是一个隐形杀手，不稳定的产品就像性格不稳定的人，完全没有安全感！要在正常的网站运行流程外，增加异常捕获机制，避免网站出现奔溃的情况。要合理设计业务流程和测试用例，提高网站的可靠性。

 （4）系统健壮性

网站运行时，需要接收的数据要保证正确性，在网站运行中在需要用户输入数据的地方，需要对用户的输入进行判断，防止一些常用的攻击手段，保证用户输入合法，尽量避免异常地出现。

4 系统的设计

4.1 系统的体系结构

系统设计是经过系统分析，然后由抽象至具体的过程，本阶段的主要目标是把在分析阶段中的抽象概念化作能够实现，使之成为有具体功能的模块，提供支持给系统代码设计。在系统设计的阶段种，应该尽可能全面的思考全系统的功能与需求，由于情况的不同，有时还要考虑到系统性能。打砖块游戏的画面简单，功能也不怎么复杂，系统资源占用很少，所以，在这里不过于考虑性能需求，下面对游戏的详细设计过程做一个说明。

游戏系统设计如图4.0

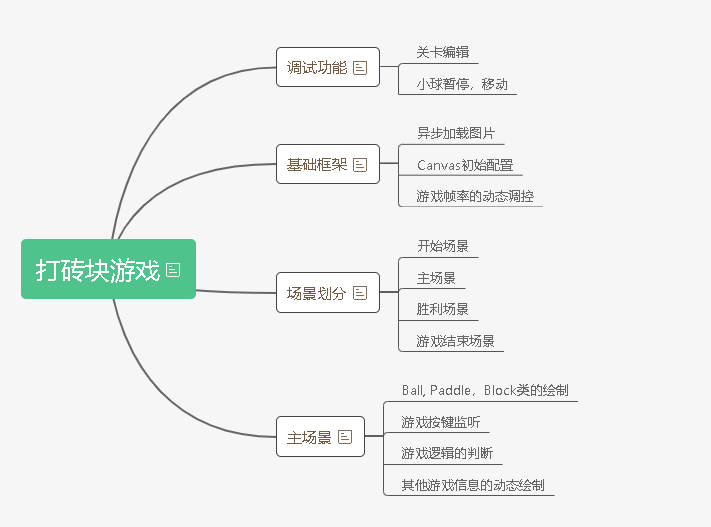


图4.0 游戏系统设计

4.1.1 游戏界面绘制模块

界面绘制模块的作用是负责游戏界面的显示。

游戏界面为一矩形窗口。

游戏界面内容：

1. 游戏信息：小球生命值显示、游戏得分、游戏等级，随着游戏不断更新；

2. 游戏功能：砖块、小球、挡板；

3. 游戏各个场景，比如开始场景，编辑场景，胜利场景，失败场景等等，当游戏开始后会隐藏。

游戏绘制界面如图4.1，

1· 

图4.1 游戏绘制

4.1.2 游戏操作模块

游戏操作模块，包括砖块的生成，挡板的生成，挡板的移动，小球的运动，消除砖块，游戏得分，游戏等级等。要特别注意小球与挡板移动后会不会超出操作区域等问题。  游戏过程中，玩家是通过键盘或鼠标控制游戏。玩家输入操作命令，程序接收并做出响应。下面列出鼠标、键盘的按键和对应的功能：

（1）A，D键控制挡板的左右移动；

（2）F键弹出小球，K箭开始游戏；

（3）R键：游戏重新开始；

（4）E键：游戏编辑界面；

（5）鼠标拖动：游戏界面能移动小球；

（6）鼠标左键：游戏编辑界面添加砖块；

（7）鼠标右键：游戏编辑界面去除砖块；

 游戏操作如图4.2，

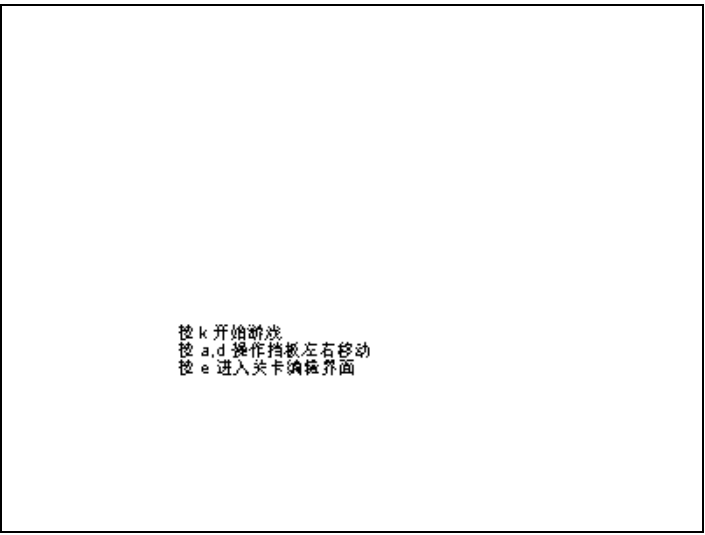


图4.2 游戏操作

4.1.3 游戏调试模块

游戏调试模块，包括调节速率，游戏暂停，移动小球等功能。

（1）点击游戏下方滚动条调节游戏速率

（2）点击小球可拖拽小球；

（3）P键：暂停游戏；

（4）自定义log窗口；

（5）自定义关卡界面；

 游戏调试如图4.3，

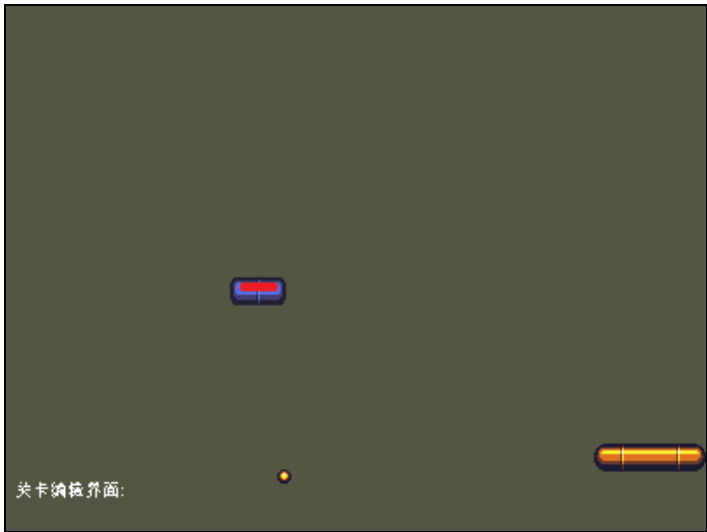


图4.3 游戏调试

5 系统的实现

5.1 游戏框架的设计

Main.js是游戏的入口文件。里面存放了一些全局函数和全局变量。

主要是得分，当前关卡，剩余生命等等。guaGmae是游戏运行的配置函数

是游戏的主入口。在guaGmae的基础上，我们在一个个scene来切分游戏。

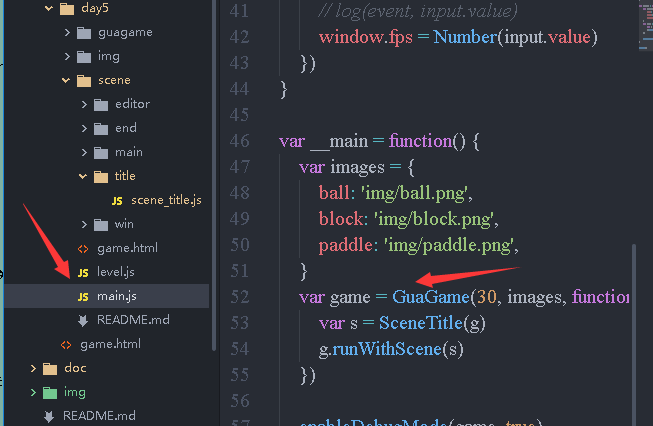


图5.1 main.js

关键代码：

var \_\_main = function() {

var images = {

ball: 'img/ball.png',

block: 'img/block.png',

paddle: 'img/paddle.png',

}

var game = GuaGame(30, images, function(g){

var s = SceneTitle(g)

g.runWithScene(s)

})

enableDebugMode(game, true)

}

\_\_main()

5.2 scene场景的实现

游戏初始化后，先从scene\_title开始，一步步依据游戏逻辑而进入main scene，win scene，end scene等等的场景 scene是游戏模块的基本单位。

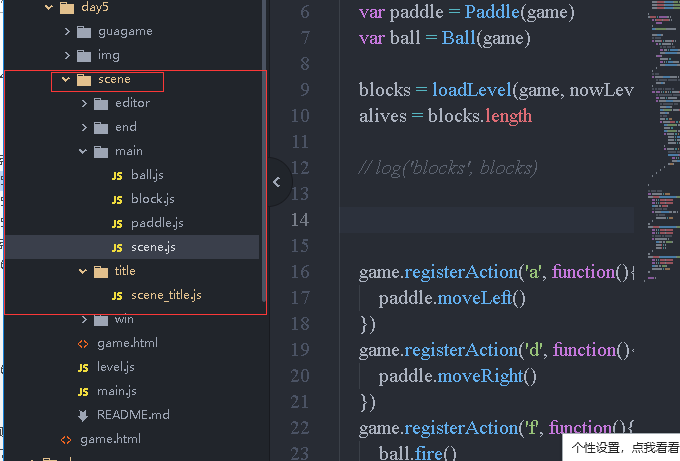


图5.2 scene

关键代码：

var Scene = function(game) {

var s = {

game: game,

}

// 初始化

var paddle = Paddle(game)

var ball = Ball(game)

blocks = loadLevel(game, nowLevel)

alives = blocks.length

// log('blocks', blocks)

game.registerAction('a', function(){

paddle.moveLeft()

})

game.registerAction('d', function(){

paddle.moveRight()

})

game.registerAction('f', function(){

ball.fire()

})

s.update = function() {

if (window.paused) {

return

}

log("nowLevel",nowLevel)

ball.move()

// 判断游戏结束

if (ball.y > paddle.y) {

if (! userLife) {

// 跳转到 游戏结束 的场景

var end = SceneEnd(game)

game.replaceScene(end)

}else{

userLife --

var s = Scene(game)

game.replaceScene(s)

}

}

// 判断相撞

if (paddle.collide(ball)) {

// 这里应该调用一个 ball.反弹() 来实现

ball.反弹()

}

// 判断 ball 和 blocks 相撞

for (var i = 0; i < blocks.length; i++) {

// log('alives', alives)

var block = blocks[i]

if (block.collide(ball)) {

// log('block 相撞')

block.kill()

log('alive', block.alive)

ball.反弹()

// 更新分数

score += 100

if (block.alive == false && alives>0 ) {

log('alives', alives)

alives --

log('alives', alives)

if (alives == 0 && blocks.length<=2) {

blocks = loadLevel(game, blocks.length + 1)

alives = blocks.length

nowLevel = blocks.length

}

if (alives == 0 && blocks.length >2) {

//跳转到you win 场景

setTimeout(function () {

var s = SceneWin(game)

game.replaceScene(s)

},500)

}

}

}

}9

}

5.3主场景控件的实现

主场景的控件主要是ball，paddle，blocks等等。分别通过var ball = Ball() 这样的形式初始化实例。实例中包含实例的一些基本变量和函数，比如在画面中的初始位置，速度，碰撞函数等等。

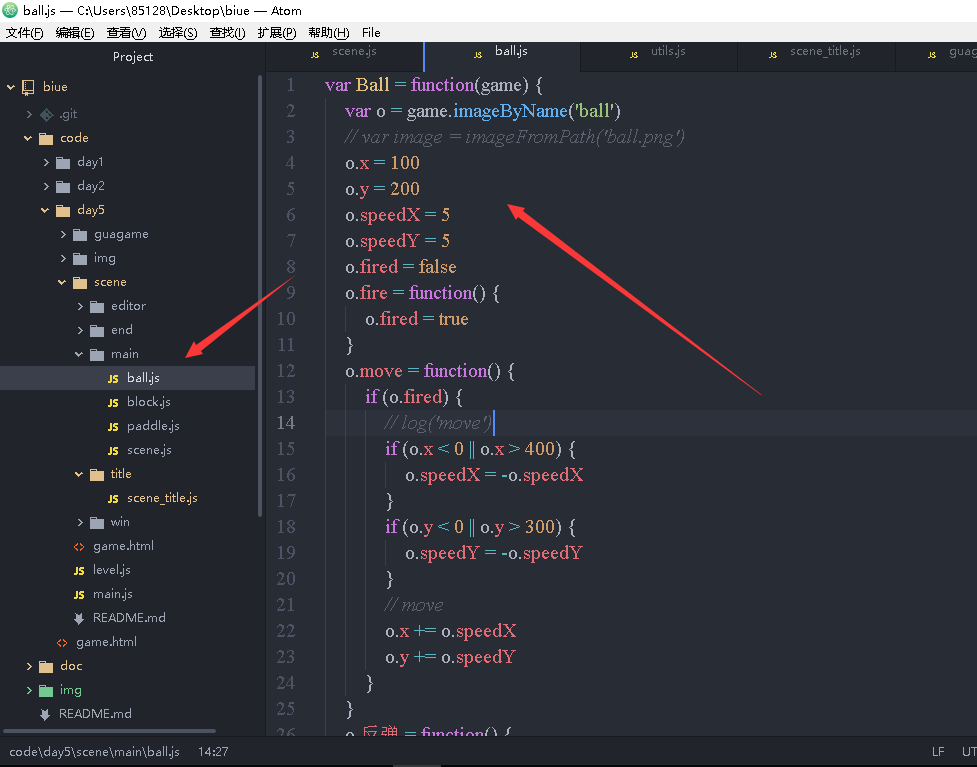


图5.3 ball.js

关键代码：

@AuthPassport(isEditor = true)

@ResponseBody

@RequestMapping("/sendNews")

public Result sendNews(String title, String content, HttpServletRequest request) {

User user = (User) request.getSession().getAttribute("User");

return newsService.sendNews(title, content, user.getId().longValue());

}

@Override

public Result sendNews(String title, String content, long userId) {

try {

NewsWithBLOBs news = new NewsWithBLOBs(title, content, userId);

String str = Utils.delHTMLTag(news.getNewsContent()).replace(" ", "").replace("　", "").trim();

news.setNewsIntro(str.length() > 150 ? str.substring(0, 150) : str);

if(newsMapper.insert(news) > 0) {

return new Result(ResultInfo.SUCCESS);

}

} catch (Exception e) {

e.printStackTrace();

}

return new Result(ResultInfo.ERROR);

}

5.4 按键监听模块实现

首先在游戏的框架类guagame里定义两个对象actions: {}, keydowns: {} 。actions里定义按键对应的callback函数。而keydowns记录按键的按下状态。最后调用全局函数window.addEventListener监听按键，改变两个对象的状态。

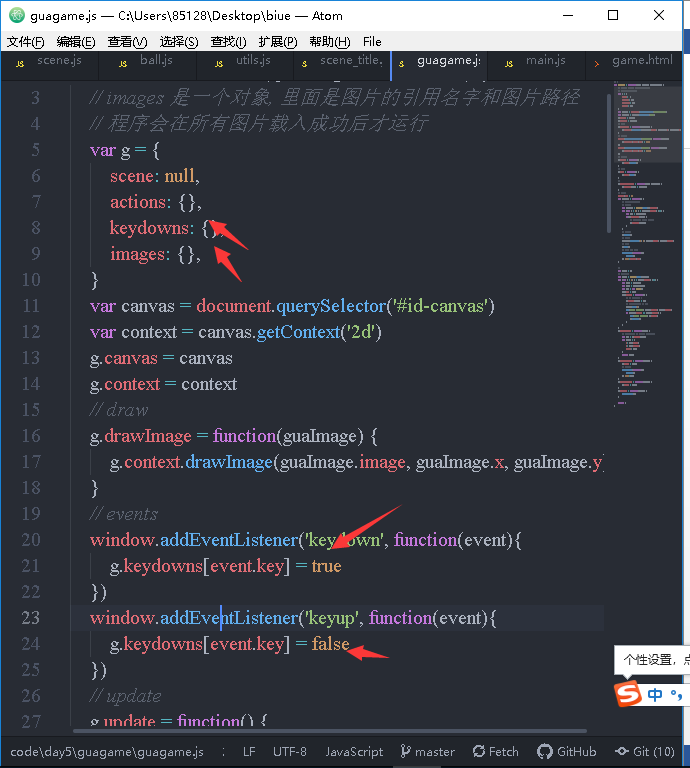


图5.4 按键监听模块

关键代码：

var GuaGame = function(fps, images, runCallback) {

// images 是一个对象, 里面是图片的引用名字和图片路径

// 程序会在所有图片载入成功后才运行

var g = {

scene: null,

actions: {},

keydowns: {},

images: {},

}

var canvas = document.querySelector('#id-canvas')

var context = canvas.getContext('2d')

g.canvas = canvas

g.context = context

// draw

g.drawImage = function(guaImage) {

g.context.drawImage(guaImage.image, guaImage.x, guaImage.y)

}

// events

window.addEventListener('keydown', function(event){

g.keydowns[event.key] = true

})

window.addEventListener('keyup', function(event){

g.keydowns[event.key] = false

})

// update

g.registerAction = function(key, callback) {

g.actions[key] = callback

5.5 关卡编辑模块实现

关卡编辑模块完完全全是我独立构思设计的模块，其核心是在于砖块落点判定，重复点击判定，以及左右按键的监听，最后还有数据的存储等等核心功能。

关键代码：

game.canvas.addEventListener('mousedown',function(event) {

var button = event.button

var x = event.offsetX

var y = event.offsetY

var needx = (parseInt(x / 40) \* 40)

var needy = (parseInt(y / 19) \* 19)

var n = 1

var p = [needx, needy, n]

if(button == 0){

if (editorBlocks.length == 0){

// log("editorBlocks.length == 0")

let block = Block(game, p)

editorBlocks.push(block)

}else {

let haddosomething = false

for (var i = 0; i < editorBlocks.length; i++) {

if (editorBlocks[i].hasPoint(x, y)) {

if (editorBlocks[i].lifes == 1) {

editorBlocks[i].lifes += 1

}

haddosomething = true

break;

}

else if (editorBlocks[i].hasPoint(x, y) == false) {

continue;

}

}

if (haddosomething == false) {

let block = Block(game, p)

editorBlocks.push(block)

haddosomething = true

}

}

log('editorBlocks', editorBlocks)

}

else if(button == 2){

let haddosomething = false

for (var i = 0; i < editorBlocks.length; i++) {

if (editorBlocks[i].hasPoint(x, y)) {

if (editorBlocks[i].lifes == 2) {

editorBlocks[i].lifes -= 1

}else if (editorBlocks[i].lifes == 1) {

editorBlocks.splice(i,1)

}

haddosomething = true

break;

}

else if (editorBlocks[i].hasPoint(x, y) == false) {

continue;

}

}

}

save()

// fetch()

})

return s

6 系统测试

6.1 测试计划

基本类测试：15天、游戏流畅度测试：8天。

6.1.1 测试进度

基本类测试：2018.01.01-2018.01.15

游戏流畅度测试：2018.02.12-2018.02.20

6.1.2 测试方案

要想开发的游戏被玩家接受与喜爱，最基本的是游戏能正确合理运行，所以，游戏的测试就变得非常重要，我想要我设计的游戏有良好的健壮性和稳定性。能在其基本的游戏上扩展，所以我主要是测试基本类是否方便扩展是一个方面，另外就只能通过简单的黑盒测试游戏本身是否与设计需求相符。最后便是我加入了一个调试模块，可以让我更方便地测试调试游戏。

6.1.3 测试目的和原则

本次测试要找出游戏的各种错误、缺陷，发现游戏中不合理的地方，进一步完善游戏内容，增强用户的体验，确保游戏正常运行。  根据测试目的，本测试的测试原则如下：

（1）测试用例的设计，必须得同时包含合理的和不合理的输入。

（2）测试用例要有输入参数与预期输出的结果两部分。

（3）测试计划必须严格单步执行。

（4）对测试结果检查和记录。

（5）为了后续游戏完善的方面，应该将测试材料归档。

（6）通过用例、错误等数据统计并分析结果。

6.1.4 测试环境

系统硬件需求：

较高主频的CPU，512MB以上内存。

系统软件需求：

Windows 7以上的操作系统、Chrome 63.0.3239.132以上的版本浏览器。

6.2 测试说明

6.2.1 游戏界面显示测试

游戏的功能测试主要对游戏初始界面、小球生命值显示、游戏等级显示、游戏调试等模块进行测试。

1. 游戏初始界面显示 输入：刷新游戏  输出：游戏初始界面，如图6.1 结果：正常



图6.1 游戏初始界面

1. 小球生命值显示 初始状态：显示剩余3条生命 输入：挡板没有挡住小球 输出：生命值显示减少，如图6.2 结果：正常



图6.2 游戏进行界面

1. 游戏等级升级显示 初始状态：等级显示为1 输入：清除第一关所有砖块 输出：等级显示为2，如图5-4 结果：正常



图6.3 Level升级界面

1. 游戏通关后界面显示 输入：通过所有关卡  输出：“You Win,Press R To Restart”，如图6.4 结果：正常



图6.4 游戏通关界面

6.2.2 挡板移动控制测试

（1）按键控制挡板

输入：分别按下“a”、“d”按键

输出：挡板按照游戏规则移动，挡板移动正常，如图6.5



图6.5挡板测试

6.2.3 小球移动控制测试

（1）小球发射检测

输入：进入游戏后按下“F” 输出：小球开始运动 结果：正常

输出：小球开始运动

结果：正常，

（2）小球碰撞检测

①测试小球与边框、挡板碰撞

输入：开始游戏观察碰撞

输出：小球碰撞后进行反弹

结果：正常

②测试小球与砖块碰撞反弹

输入：开始游戏观察碰撞

输出：小球与砖块碰撞后，消除砖块并正常反弹

结果：正常

如图6.6



图6.6小球碰撞

7 测试及性能分析

7.1 测试结果概述

测试过程中，程序出现了一些错误，界面不能正常加载、小球不能移动、操作命令无响应等，测试和调试后，解决了这些问题，完成了这个可以正常运行的版本，当前游戏的开发结束。

7.2 测试结果评价

经测试，系统功能需求的核心功能都已经实现，并且运行良好。

8 总结与展望

8.1 本论文研究成果

通过此次毕业设计，我更深入的了解了H5，Canvas,而且能用他做一些有趣的小游戏。之前学过JSP，但是没用它做过游戏，这次设计使我从另一个方面了解了JSP，也是对学习的进一步加深。  打砖块游戏的开发，我认为最主要的是一些游戏算法的设计和游戏架构的设计，比如游戏几个基本元素类的设计，或者说所有的小游戏都是以这些为核心。JSP语言ES6的语法是一种面向对象的语言，所以也该考虑代码的服用，善用函数式编程，能让我们更好地完成项目，打砖块小游戏的算法本身实现并不困难。当然，游戏设计实现的过程中还是遇到了很多困难，因为以前没有用JSP做过游戏，所以很多方面不熟悉。  从最初学习JSP语法，到现在完成简单的打砖块游戏，我感觉到了自己在进步，学习和编程的能力都有很大的增强，同时，这也增加了我对游戏项目的实战经验和技巧，相信以后可以更轻松的完成此类项目。  在做项目这段时间里，我学到了许多新知识，同时复习并巩固了之前学习的旧知识，通过对它们的实践，我对我自己的专业软件工程也有了更深刻的理解。一个打砖块游戏的开发，贯穿了整个软件工程的过程，从开始的项目选定，到需求分析，到算法的设计和实现，直到最后的软件测试，把整个大学里学习的很多专业知识综合起来，构成了一个完整的体系。

参考文献

[1] Peter Lubbers[美]，Brian Albers [美]，Frank Salim[美][著]，李杰，柳靖，刘淼[译].HTML5高级程序设计[M]. 北京，人民邮电出版社，2011.

[2] ChristianHeilmann[德][著].牛海彬[译]. 深入浅出JavaScript[M]. 北京,人民邮电出版社,2008.

[3] Rob Hawkes[英][著]，周广新，曾少宁，盛海艳[译].HTML5 Canvas基础教程[M]. 北京，人民邮电出版社, 2012.1.

[4] 百度百科.打砖块.http://baike.baidu.com/view/28513.htm.

[5] 罗彪,张宏涛.基于HTML5的移动互联网应用发展趋势[J].信息与电脑(理论版),2018(10):129-131.

[6] 和歆雨.基于HTML5的网页游戏的设计与开发[J].中国战略新兴产业,2018(04):94-95.

[7] 史济民，顾春华，郑红.软件工程:原理、方法与应用[M].北京:高等教育出版社,2009:33-68.

[8] 网页设计教程/张月玲，韩毅娜主编.——北京：清华大学出版社;北京交通大学出版社,2006.8.

[9] Web程序设计——JSP/孙延鹏,吕晓鹏编著.——北京:人民邮电出版社,2008.5.

[10] IT项目管理:从理论到实践/王如龙主编.——北京:清华大学出版社,2008.9.

致 谢

在毕业论文完成之际，我要特别感谢邓邵伟导师的热情关怀和悉心指导。撰写毕业论文以及修改项目的过程中，邓邵伟导师倾注了大量的心血和汗水，无论是在毕业论文选题、构思和资料的收集方面，还是在毕业论文的研究方法以及成文定稿方面，笔者都得到了邓邵伟导师悉心细致的教诲和热情无私的帮助。本次毕业设计的每一个地方都凝聚着邓老师的心血，没有邓老师的悉心指导和帮助，就没有本论文的顺利完成和我的成长。

衷心感谢邓老师所给予的无私帮助，他渊博的学识、敏锐而超前的学术意识、民主而严谨的治学作风将使我受益终生，并激励我在今后的工作、学习和生活中，时刻保持勤奋刻苦的工作态度和求实创新的敬业精神。

最后，向在百忙之中抽出时间对本文进行评审并提出宝贵意见的各位老师表示衷心的感谢！