

PaperPass检测报告简明打印版

比对结果（相似度）：

总体：7 %（总体相似度是指本地库、互联网的综合比对结果）

本地库：5 %（本地库相似度是指论文与学术期刊、学位论文、会议论文数据库的比对结果）

互联网：2 %（互联网相似度是指论文与互联网资源的比对结果）

编号：5752D8A81CC5E916M

标题：基于cocos2d-lua的五子棋设计与实现

作者：刘博

长度：15091 字符(不计空格)

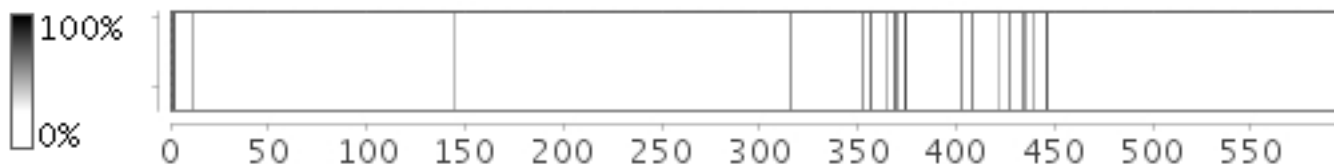
句子数：595句

时间：2016-6-4 21:33:28

比对库：学术期刊、学位论文（硕博库）、会议论文、互联网资源

查真伪：<http://www.paperpass.com/check>

句子相似度分布图：



本地库相似资源列表（学术期刊、学位论文、会议论文）：

- 相似度：3 % 篇名：《基于扩展Alpha-Beta算法的中国象棋软件实现的研究》
来源：学位论文 东北大学 2006 作者：白硕
- 相似度：2 % 篇名：《六子棋计算机博弈关键技术研究》
来源：学位论文 重庆交通大学 2010 作者：闵文杰
- 相似度：2 % 篇名：《计算机五子棋博弈系统的研究与实现》
来源：学位论文 山东师范大学 2005 作者：董红安
- 相似度：2 % 篇名：《计算机博弈的估值方法研究》
来源：学位论文 东北大学 2008 作者：刘明慧
- 相似度：1 % 篇名：《五子棋中Alpha-Beta搜索算法的研究与改进》
来源：学术期刊 《计算机工程》 2012年17期 作者：程宇 雷小锋
- 相似度：1 % 篇名：《具有自学习功能的计算机象棋博弈系统的研究与实现》
来源：学位论文 哈尔滨工程大学 2007 作者：王一非
- 相似度：1 % 篇名：《基于WindowsCE平台的游戏博弈系统研究与实现》
来源：学位论文 东北大学 2008 作者：陈威
- 相似度：1 % 篇名：《四国军棋的人机博弈搜索算法研究》
来源：学位论文 南京航空航天大学 2008 作者：陆慧

9. 相似度：1 % 篇名：《基于Intel多核多处理器的两个经典游戏AI算法的并行化研究》

来源：学位论文 东南大学 2010 作者：邹峰

10. 相似度：1 % 篇名：《机器博弈中搜索算法的研究》

来源：学术期刊 《福建电脑》 2012年10期 作者：廖景亮 陈冬强

11. 相似度：1 % 篇名：《哈希技术在中国象棋机器博弈系统中的应用研究》

来源：学术期刊 《科学技术与工程》 2008年17期 作者：高强 郭琛

12. 相似度：1 % 篇名：《五子棋人工智能算法实现研究》

来源：学术期刊 《散文百家（下）》 2014年7期 作者：段晓丽

13. 相似度：1 % 篇名：《智能五子棋博弈算法研究》

来源：学术期刊 《江苏技术师范学院学报》 2013年2期 作者：王云霞

14. 相似度：1 % 篇名：《混合博弈树算法在中国象棋人机博弈中的应用研究》

来源：学位论文 大连交通大学 2008 作者：鹿荣

15. 相似度：1 % 篇名：《基于Brew平台的人工智能的研究》

来源：学位论文 北京邮电大学 2006 作者：许云春

16. 相似度：1 % 篇名：《棋类博弈算法的改进》

来源：学术期刊 《现代电子技术》 2005年1期 作者：瞿锡泉 白振兴 包建平

17. 相似度：1 % 篇名：《博弈及其常用搜索算法初探》

来源：学术期刊 《无线互联科技》 2011年12期 作者：戴翔

18. 相似度：1 % 篇名：《博弈树搜索算法概述》

来源：学术期刊 《计算机系统应用》 2009年9期 作者：岳金朋 冯速

19. 相似度：1 % 篇名：《基于alpha-beta剪枝搜索算法的中国象棋游戏设计》

来源：学术期刊 《信息通信》 2015年8期 作者：刘淑英 穆远彪 李红

20. 相似度：1 % 篇名：《智能五子棋博弈程序的核心算法》

来源：学术期刊 《枣庄学院学报》 2005年2期 作者：董红安 蒋秀英

21. 相似度：1 % 篇名：《连珠模式棋类博弈的搜索优化》

来源：学位论文 安徽大学 2014 作者：王小龙

22. 相似度：1 % 篇名：《四国军棋人机博弈系统及有约束推理的研究和实现》

来源：学位论文 南京航空航天大学 2010 作者：张红明

23. 相似度：1 % 篇名：《基于局势变化的计算机中国象棋研究》

来源：学位论文 东北大学 2008 作者：王友政

24. 相似度：1 % 篇名：《基于置换表技术的中国象棋搜索算法研究》

来源：学术期刊 《科技视界》 2014年16期 作者：黎利辉

互联网相似资源列表：

1. 相似度：2 % 标题：《棋类游戏的智能模块设计 浅谈人工智能 学步园》

<http://www.xuebuyuan.com/583529.html>

2. 相似度：1 % 标题：《Android之Wifi学习(1) - 火山哥的专栏 ...》

<http://blog.csdn.net/sunchaoenter/article/details/6593241>

全文简明报告：

{ 100 % : 辽宁科技大学本科生毕业设计（论文）第1页 }

基于cocos2d-lua的五子棋设计与实现

摘要

现如今我们已然步入21世纪，而移动互联信息技术也已进入了高速发展的阶段，并且使得智能手机在世界上得到了大范围的普及，各式各样，五花八门的手机应用也应运而生，智能手机也因为手机应用的丰富多彩而更加深入人心。手机应用分为多种，旅游出行、金融理财、视频、购物、音乐、图像、生活实用工具、游戏等等。
{ 42 % : 在众多的手机应用中，手机游戏恰恰是伴随手机发展历史最久的手机应用。 } 在百花齐放的手机应用市场中手机游戏依旧是最热门的下载选择，原因无他只因为现如今社会节奏日益加快，生活压力也日渐增多，在智能手机大范围普及的今天，手机无疑是最方便快捷的娱乐工具， { 70 % : 而手机游戏便是它娱乐性的最佳体现应用。 } 本款五子棋手机游戏正是迎合着现如今智能手机用户的娱乐需求应运而生，其玩法简单，老幼皆宜。 { 45 % : 游戏功能丰富，可人机对弈、双人对弈、蓝牙联对弈、wifi联机对弈。 } 本款游戏不仅完全体现出游戏的娱乐性，而且五子棋本身就是一种益智的棋类竞技游戏，经常玩五子棋可锻炼人的数学能力、智力，让玩家变得更加聪明，对逻辑思维和耐心有明显的提高。该系统主要采用cocos2d-x + lua编写，蓝牙和wifi等功能需要调用android或者ios的接口，这需要ios和android的技术，其中还涉及到JNI和人机博弈算法等知识。该游戏因为采用cocos编写，可部署在android、ios、win32、mac、linux等多个平台上，解决了平台的约束性和分平台编写代码的困扰。下面将详细的讲解本游戏的功能及实现。

关键词: 五子棋; cocos2d-x; lua; ios; android; 蓝牙; wifi; 人工智能

The design and Implementation of the Gobang based on the cocos2d-lua

ABSTRACT

With the rapid development of the Internet Technology in the 21 century, the demand for the telephone is increasing with a surprising speed, which leads to the creation of diverse kinds of mobile software to satisfied the consumers. To be specific, there are many kinds of software such as those for traveling, financing, video-watching, shopping, music-listening and so on. However, the software of game is the most profitable one among all the apps because of the entertainment it brings for users. By this I mean, along with the improvement of the society, people from all walks of life are under too much pressure for different reasons, which create a big demand for the amusement to release their emotions. However, mobile phone as the most convenience equipment becomes the first selection for the most people to play with. Under this background, this thesis will provide a design for a phone game- The Gobang. This game is made under the demand of the users in the modern society with abundant game moderns: single battle, double battle, battle by the blue-tooth, battle via wifi. This game can not only present the amusement of phone-games, but also enhance the intelligence of the players. Therefore, those who usually play the game will have an absolutely increasing of their logistics and patience. The system is mainly write by the cocos2 d- x+ lua, bluetooth, as well as the wifi which requires the technique of ios and android, and also need the knowledge of JNI and the algorithm. Due to the fact that this game is designed by the cocos, which means it can be applied on the android, ios, win32, mac, linux and so on and thus create a large developing space. Then, this thesis will give a specific description for the game.

Key words: gobang; cocos2d-x; lua; ios; android; bluetooth; wifi; AI

目录

TOC \o "1-3" \u1 绪论1

1.1 课题背景1

1.1.1研究背景1

1.1.2国内外现状2

1.2 研究目的4

1.2.1系统的必要性4

1.2.2系统的可行性4

1.3 实现系统的技术介绍5

1.3.1 cocos2d-x介绍5

1.3.2 C++语言介绍5

1.3.3 lua语言介绍5

1.3.4 java语言介绍6

1.3.5 objective-c介绍6

1.3.6 JNI介绍6

1.3.7 蓝牙技术介绍7

1.3.8 wifi技术介绍7

1.3.9 人工智能介绍7

2 需求分析8

2.1 系统目的背景及意义8

2.2 功能需求8

2.2.1 界面需求8

2.2.2 玩法需求9

2.2.3 性能需求9

2.3 系统用例图10

2.4 运行环境10

3 系统设计10

3.1概要设计11

3.1.1 游戏主体设计11

3.1.2 联机模块设计13

3.2详细设计14

3.2.1游戏流程设计14

3.2.2 棋盘类设计16

3.2.3 游戏基类设计17

3.2.4 双人对弈设计17

3.2.5人机对弈设计17

3.2.6蓝牙对弈设计25

3.2.7 wifi对弈设计25

3.2.8 结束页设计25

4 系统实现26

4.1 工程搭建27

4.1.1 环境搭建27

4.1.2 创建工程28

4.2 游戏首页实现29

4.3 游戏规则界面实现29

4.4 游戏主界面实现29

4.5 游戏结束界面实现30

4.4 双人对弈实现31

4.5 人机对弈实现31

4.6 蓝牙对弈实现31

4.7 wifi对弈实现33

结 论33

致 谢34

参考文献35

2 需求分析

2.系统目的背景及意义

{ 47 % : 智能手机也像电脑一样，有着自己的操作系统，现在的手机系统主要有苹果的IOS、谷歌的android和微软的Windows Phone三大流行系统。 } 现在的智能手机游戏也有其主攻的手机系统，就算是开源扩展市场也会是根据另一种手机系统而后将游戏做出这种游戏版本，对于游戏的开发商来说增加了不小的负担，而玩家的游戏体验由于受到了系统平台的约束也是大打折扣。而本款五子棋游戏是应用coco2d-x游戏引擎进行跨平台游戏开发，消除了玩家游戏约束，减少了游戏开发商的负担。

同时，我国手机游戏还在快速发展的今天，如何将充分发挥手机游戏的性能已然成为了游戏开发商，运营商以及玩家们心中最为至关重要的问题。功能全面，轻松耐玩，富有创意的手机游戏会使得玩家在对该手机游戏产品的喜爱及认可的同时也为手机游戏开发商与运营商带来与其付出成正比的经济收入。对于本款手机游戏五子棋，玩家可以通过一部手机同好友一起进行游戏，若是在一定区域内的两个陌生人手机里都存有本款游戏的话，还可以通过蓝牙或者wifi进行连接对战，在享受游戏快乐的同时还可以结识一个志同道合的棋友；而且本款手机游戏对于之前并不了解亦或者不擅于五子棋的玩家也可以起到教学和考练的功能。原因无他，只因为本款游戏具有提示功能，根据五子棋的程序算法来提示玩家落子方位，以方便玩家进行游戏，并可以让玩家习得五子棋的法门，更增添了一份窝心的趣味。

2.2 功能需求

2.2.1 界面需求

{ 46 % : 界面分为首页，游戏规则界面，游戏主页面和游戏结束页面。 }

1.游戏首页： 需要显示游戏名称（五子棋），需要提供玩法入口按钮，分别为人机对弈、双人对弈、蓝牙对

弈、Wi-Fi对弈。 还要有游戏规则入口按钮。

2.游戏规则界面： 需要横向切换游戏规则讲解界面，每个讲解界面需要提供一个跳过的按钮。 {49%：讲解界面要分别介绍棋盘、五子连珠、活三、活四、冲四、长连、} {48%：禁手、三三禁手、四四禁手、行棋落子等五子棋相关概念。}

3.游戏主页面： 需要有15*15的棋盘，游戏时间显示，当前下子方显示，返回游戏首页、声音、重玩、悔棋、提示、求助等菜单。 当点击声音时出现一个下拉菜单，提供音乐和音效的开关按钮。

4.游戏结束页面： 需要显示赢棋方（黑棋或着白棋），布局时间、本局步数的显示，还要提供悔棋，重新挑战，返回主页等功能按钮。

2.2.2 玩法需求

1.人机对弈： 人机轮流下子，当电脑下子时玩家不能下子，显示电脑正在下子状态。 悔棋时需要悔两步子，即悔掉电脑下的子和玩家下的子，然后玩家在重新下子。 首次玩时默认是电脑先下子，其后轮流先下子。

2.双人对弈： 两个人在一个手机上下五子棋，轮流下子，悔棋时悔一步，即悔掉当前玩家下的子，然后重新下子。

3.蓝牙对弈： {61%：当点击蓝牙对弈时，进入游戏界面。} 如果玩家的手机是ios系统，当用户手机没开启蓝牙，提示开启蓝牙，然后提示正在搜索设备。 如果用户手机蓝牙开着，则直接进入搜索界面。 当设备成功配对后，玩家可下子，对方的下子状态在己方的手机上实时显示。 如果是玩家的手机是 android系统，当玩家选择蓝牙对弈进入游戏界面后，提示玩家是否允许蓝牙在300秒内可见， 当玩家选择是时弹出创建游戏和加入游戏的选择对话框，玩家选择创建游戏则弹出 waiting对话框，当玩家选择加入游戏时弹出以搜索到的设备列表， 还可扫描设备，玩家可在设备列表中选择对应的设备加入游戏。 当玩家提出悔棋，重玩等操作时，另一个玩家应收到请求通知，经过另一个玩家同意，方可悔棋或者重玩，并且棋盘状态同步更新。 当点击提示时，弹出友好提示，告知提示功能不可用。

4. wifi对弈： 当点击wifi对弈时，列出wifi列表，玩家可以加入同一个wifi，然后显示同一wifi下的玩家。 玩家们可以邀请那些没有加入游戏的玩家和自己对局。 玩家还可以选择创建热点的方式来玩游戏。 悔棋重玩等游戏的交互可参考蓝牙对弈。

5.其他： 需提供悔棋，重玩，提示，求助等功能。 当盘面上五子连珠时显示结束界面，显示出赢棋方。在五子连珠后玩家不可下子，但可选择悔棋，再玩一次和返回首页等操作。

2.2.3 性能需求

1.经研究表明，人的等待时间在3秒内为宜，超出3秒后，人会变得焦躁。 因此本游戏的人机对战环节要求AI下棋时间应该在3秒内。

2.蓝牙和wifi的延时不能过长，若延时过长，影响游戏的体验性，1秒之内为宜。 经测试，蓝牙的延时为300~500毫秒左右，wifi延时为280毫秒左右，符合该条件。

2.3 系统用例图

玩家进入游戏界面，游戏首页展现4种玩法，人机对弈，双人对弈，蓝牙对弈，wifi对弈。 相关的游戏操作又有悔棋，重玩，提示，求助等。 其系统用例图如图2.1：

图2.1 系统用例图

2.4 运行环境

1.开发环境： 该游戏采用xcode、sublime-text，android-studio等编译器，游戏引擎为cocos-lua3.5版本。 其代码采用git管理。

2.运行环境： 该游戏可运行在ios和android4.0即4.0以上设备。

3.项目限制： 蓝牙对弈时两个手机距离不能过远，10米内为宜，最好不要有遮挡物。 wifi对弈时两个手机要处在同一局域网下。

3 系统设计

3.1概要设计

3.1.1 游戏主体设计

本游戏因为设计到多种玩法，每个玩法里都有些共同的功能，如下棋、重玩、悔棋、提示、求助、时间显示、当前下子方、返回游戏首页、游戏结束页等。 所以应该把这些公共的共能抽出来，由一个基类来管理，本游戏这个基类为GameBaseScene，该类提供了悔棋、重玩、求助、提示、时间显示、当前下子方显示等功能。 因此只要具体玩法继承这个类，便有了这些共能。 其下子、悔棋、重玩、判断游戏是否结束等具体的功能实现在ChessboardNode这个类中，这个类负责管理棋盘的相关操作。 从某种意义上来说，下子的逻辑与游戏主体实现了解偶。 而人机对战部分，其相关的人机对战算法较为通用，可摘取出来以便在其他处使用， 因此可分装个AI，提供人机对战算法，传进当前棋盘状态即可返回电脑下子的最佳位置。 TranspositionTable为置换表，是为了优化人机算法的运行速度的。 相关的类图设计图下：

1.棋盘类

图3.1 棋盘类图

2.游戏基类

图3.2 游戏基类图

3. AI类

图3.3 AI类图

4. 置换表

图3.4 置换表类图

5. Gameover类

图3.5 游戏结算类图

该项目类之间的关系如图3.6

图3.6 类关系图

3.1.2 联机模块设计

本游戏提供了蓝牙对弈和wifi对弈两种联机玩法，但其需要的接口是一样的，无非就是扫描设备并进行链接、传递数据、接受数据、断开链接等。因此可把这些同样的功能抽象出来，封装个接口。然后采用静态工厂方法来生产具体的连接方式，蓝牙连接或者wifi连接。因为本游戏涉及到跨平台，因此蓝牙功能和wifi功能需要分平台实现，但单例模式是两个平台公共的，不需要分平台实现，所以在两个平台之上，又封装了个WifiManager和BleManager。提供getInstance()方法，以便工厂生产两个对象。然后涉及到连接设备，发送数据，接受数据等具体的方法，需要分平台实现。具体联机模块的设计请见图3.7

图3.7 联机模块类关系图

3.2详细设计

3.2.1游戏流程设计

本游戏可划分为游戏首页、人机对战、双人对战、蓝牙对战、wifi对战、返回首页和游戏结束页等模块，其相关的流程图如下：

1.双人对弈

图3.8 双人对弈流程图

2.人机对弈

图3.9 人机对弈流程图

3.蓝牙对弈及wifi对弈

图3.10 蓝牙或wifi对弈流程图

3.2.2 棋盘类设计

1.棋盘表示

五子棋的标准棋牌为15*15的格子，我们要采用个数组来记录每个格子的状态： 黑子、白子、空子。

2.监听棋盘点击

当点击棋盘时，计算出玩家点击的是那个格子。 然后执行外部注入的回调函数。 如下子的回调函数

```
if ( 点击棋盘 ) {  
  
    executeCallback ( row , col );
```

3.下子

若收到的下子坐标超出棋盘边界，或者游戏结束还下子，又或者该下子位置以有子，则不做任何反应，若下子坐标合理则更新棋盘数组的状态，然后在相应的棋盘位置下子。 此处我们应该设计个接口，让外部来设置先手下子的类型，默认为黑子。 下子后更新当前下子方。

4.检查是否五子连珠

当下子后，我们应该检查改子周围横、竖、左斜、右斜即每一条线上的同类棋子不间断出现的数目是否大于等于5，如果大于等于5，则游戏结束。 返回连成5子的棋子类型，即赢棋方。 如果棋盘下满时，要返回和棋的信息。

5.悔棋

当棋盘无子时不能悔棋，当棋盘有子时，每悔一次棋，要对应更新棋盘数组状态。

6.重玩

清空棋盘，初始化棋盘数组。

7.求助

将当前的棋盘状态截图分享。 涉及到的分享功能需要跨平台实现，这里可采用第三方工具ShareSdk。

3.2.3 游戏基类设计

1.加载棋盘，将棋盘对象设为protected，以便子类使用棋盘类的功能。

2.提供重玩、悔棋、提示、求助、返回首页等功能菜单，点击功能菜单时调用棋盘类实现对应的功能。

3.记录游戏时间，点击棋盘下子则认为游戏开始，直到计时到游戏结束。 游戏开始事件和结束事件可在棋盘类中注册个回调函数，当首次点击棋盘时，执行回调函数，开始计时。 棋盘上有五子连珠或者和棋时收到游戏结

束通知，停止计时器，调出游戏结束层。

3.2.4 双人对弈设计

注册点击棋盘下子的回调函数即可。

3.2.5 人机对弈设计

1. 人机博弈算法

(1) 设计个评分表，来评估某个棋型的分数，如五连给出的得分为50000分。 具体的评分规则如图3.11。

图3.11 棋型分数图

图中的圆圈代表棋子，加号为可下子，无加号出代表对方棋子，如括号7: oooo+ 表示的是 { 对方, 己方, 己方, 己方, 己方, 空棋 }。

(2) 评估某一点分数

分析该点的左斜、右斜、横向、纵向棋子类型，并将其存到表中，表结构如下

```
local chessLineRecord = {
```

```
Left = {},
```

```
Right = {},
```

```
Horizon = {},
```

```
Vertical = {}
```

然后遍历这个表，查看这个表中是否有符合上面评分表中对应的模型，如果符合对应的模型，则加上该模型分数。 遍历结束后返回一个总分，即为该点的估分。

(3) 评估棋盘分数

评估棋盘分数分两种，一种是对空位评分，一种是对局面评分。

a. 空位评分： 遍历可下子的空位，分别模拟人和电脑进行下子，然后分别取电脑下子和人下子的最大分数，返回最大分数之差。 这个差值是电脑得分减去人得分，还是人得分减去电脑得分，还要根据后面的具体算法来定。

b. 局面评分： 遍历以下子点，评估该点分数，如果该点是电脑下的子，则将该点分数记为电脑得分，如果是人下的子，则记为人的得分。 最后返回最大得分之差。 这个差值是电脑得分减去人得分，还是人得分减去电脑

得分，还要根据后面的具体算法来定。

(4)找出可下子点

为了减少搜索遍历的节点数，加快搜索速度，我们这里认为某空位周围两格之内有棋子，才考虑在该点下子。遍历棋盘，找出这样的空位，存进数组中，为后续的搜索提供可下子点。

(5)简易五子棋算法

该算法的思想是找出棋盘得分最大的点，然后返回该点作为下子点。做法为遍历可下子点，然后模拟人和电脑分别在该位置下子，取该位置人的得分和电脑的得分。遍历之后，得出人最大的得分和电脑最大的得分。如果人的最大得分大于电脑的最大得分，则返回人最大得分点作为下子点，否则返回电脑最大得分点作为下子点。

(6)极大极小算法

这是一个前瞻n步的算法，模拟玩家和电脑轮流下n步子，然后返回对电脑利益最大的下子点。而实际上，每走一步都对应了m种走法，把这些走法展开可形成一棵巨大的博弈树，如图3.12。

图3.12 博弈树

{ 44 % : 这个博弈树由于兄弟节点过多，不能采用宽度优先搜索算法，应采用深度优先搜索算法。 } 我们假设 A层为电脑落子点，B成为玩家落子点，棋盘估值函数返回的是电脑得分减去玩家得分，那么 A层应该选择得分最大的点走棋，B层应该选择得分最小的点走棋。而每一层的得分都是由子节点返回的，

如图3.13，即搜索到预先设置的深度时，评估当前棋盘得分，如果是玩家下子层，取子节点中最小的得分最为该点得分， { 52 % : 如果是电脑下子层，则取子节点中最大的得分作为该点得分。 } 这样遍历棋盘，在每个可下子点前瞻n步，得到该点得分，遍历可下子点后，取得最大得分点即为电脑下子点。

图3.13 极大极小算法

伪代码如下：

```
int MinMax(局面 p , int depth){ //depth是搜索深度
```

```
int bestvalue , value;
```

```
if(isGameOver || depth==0){ //叶子节点
```

```
    返回估值(p); //直接返回对局面的估值
```

```
if(当前是计算机走棋){
```

```
    bestvalue=-INF; //初始最佳值设为负无穷
```

```
else {  
  
    bestvalue=INF; // 初始最佳值设为正无穷  
  
    { 95 % : for(每一个合法的走法){//走法的生成与具体问题紧密相关，具体方法省略}  
  
        走一步棋; //局面p随之改变  
  
        value=MinMax(p, depth-1); //搜索子节点  
  
        撤销刚才的一步; //恢复局面p  
  
        if(当前是计算机走棋){  
  
            bestvalue = max(value, bestvalue); //取最大值  
  
        } else {  
  
            bestvalue = min(value, bestvalue); //取最大值  
  
        }  
  
        return bestvalue;  
    }
```

(7)负极大值算法

{ 52 % : 负极大值算法是对极大极小值代码上的优化，在搜索效率上没有明显的提高。 } 极大极小值算法需要根据下棋方来判断是取极大值还是极小值，其实我们可以让估值函数根据下子方来估值，当下子方为电脑时，返回电脑最大得分减去人最大得分，当下子方为人时，返回人最大得分减去电脑最大得分。然后每层选取对自己利益最大的得分即子节点的负极大值得分。相关的估值函数伪代码如下：

```
eval ( 盘面 , 下子方 ) {  
  
    if ( 下子方 == human ) {  
  
        return maxHumanScore - maxComputerScore ;  
  
    } else if(下子方 == computer){  
  
        return maxComputerScore - maxHumanScore;  
    }
```

由于回溯的特性，如果只节点返回的是maxHumanScore - maxComputerScore即对人利益最大的分数，那么电脑应该选取maxComputerScore - maxHumanScore最大的分数，即-(maxHumanScore - maxComputerScore)所以取的是子节点负极大值节点。

负极大值算法伪代码如下：

{ 95 % : //负极大中的估值却是对走棋方敏感的，因此函数参数中需要有一个走棋方的参数 }

long NegaMax(局面 p , ing Side , int depth){//depth是搜索深度

int bestvalue , value;

if(isGameOver || depth==0){//叶子节点

返回估值(p , Side); //直接返回对局面的估值

bestvalue=-INF; //初始最佳值设为负无穷

{ 95 % : for(每一个合法的走法){//走法的生成与具体问题紧密相关，具体方法省略 }

走一步棋; //局面p随之改变

value= - NegaMax(p , opSide , depth-1); { 100 % : //搜索子节点，注意前面的负号，opSide是对手 }

撤销刚才的一步; //恢复局面p

if(value>bestvalue){//取最大值

bestvalue=value;

return bestvalue;

(8)alpha-beta剪枝

{ 55 % : 极大极小算法存在一定数据冗余，举个例子，在五子棋博弈过程中，如果某一节点轮到甲走棋， } {
51 % : 而甲向下搜索节点时发现第一个节点就可以五子连珠，那么剩下的节点就不需要搜索了。 } 甲的值就是第
一个节点的值，这样就可以省去大量冗余的节点搜索。 其抛弃冗余节点的过程如图3.14 :

图3.14 alpha-beta剪枝示例图

{ 82 % : 先看左图，节点下面的数字为该点的值，节点B为18，节点D为16，由此我们可以判断节点C的值小于
等于16（取极小值）。 } { 53 % : 而节点A的值max（B，C）为18，也就是说再搜索E和F节点已经没意义了，所以
可将E、F节点省去，这样将E、F节点减去称为Alpha剪枝。 }

{ 54 % : 再看右图，B节点的值8，D节点的值18，那么C节点的值大于等于18（取极大节点）， } 而A节
点的值 min（8，18）为8，也就是说不需要C节点的其他节点， 如E和F节点的值就可以得出A节点的值。 { 92
% : 这样将D节点的后继兄弟节点减去称为Beta剪枝。 }

{ 64 % : 将Alpha和Beta剪枝应用到极大极小搜索中，就得到了alpha - beta搜索算法，其相关的伪代码如下： }


```
int alphBeta(dept , alpha , beta){

    if(gameOver)return evaluation; //胜负已分，返回估值

    if(dept[=0)return evaluation; //叶子节点，返回估值

    if(is min node){//极小节点

        for(each possible move m){//对每一可能的走法m

            make move m;

            score = alphBeta(dept 1 , alpha , beta);

            unmake move m;

            if(score [ beta){

                beta = score;

                if(alpha ] beta) return alpha;

                return beta; //返回极小值

            }else{//取极大节点

                for(each possible move m){//对每一可能的走法m

                    make move m;

                    score = alphBeta(dept 1 , alpha , beta);

                    unmake move m;

                    if(score ] alpha){

                        alpha = score;

                        if(alpha ] beta) return beta;

                        return alpha; //返回极大值
```

(9) 置换表

在搜索的过程中，往往会遇到相同的局面，这些局面在前面的搜索中已经搜索过了，我们可以直接利用搜索过的解果而不是在重新搜索遍， { 46 % : 有了这样的想法，如果我们要利用搜索过的局面，那么我们需要一张表把搜索过的局面存放在表里， } { 82 % : 然后在后续的搜索中，查看记录在表中的这些结果，果如某个节点已有记录，就直接利用记录下来的结果， } 这种方法叫置换表。

a. 置换表的表结构如下：

```
struct hashItem{  
  
    int64 checksum; { 86 % : //64为哈希值，用以验证表中数据是否是要找的局面 }  
  
    int depth; //该表项求值时的搜索深度  
  
    enum{exact , lower_bound , upper_bound} entry_type; //准确值、最差值、最好值  
  
    double eval; //代表该点的值  
  
}hashTable[HASH_TABLE_SIZE]//定义大小为HASH_TABLE_SIZE的哈希数组
```

b. 利用置换表的伪代码如下：

```
int alphaBeta(dept , alpha , beta){  
  
    value = lookupTT(dept , index); //查询置换表  
  
    if(value is valid) return value; //置换表中有该点，直接返回该点的值  
  
    { 88 % : //没查到，进行alpha beta搜索，求出节点的值value }  
  
    Search with alphaBeta  
  
    storeToTT(dept , value , index);  
  
    return value;
```

(10) 迭代加深

{ 78 % : 迭代加深是一个不断求精的过程，由低到高的一系列搜索组成。 } 用以控制搜索的时间。 例如我们要将搜索时间控制在60秒左右，要搜索尽量深的层次，伪代码表达如下：

```
while ( i [ maxDepth ) { //i为搜索深度，初值为0
```

```
alphaBeta(i , alpha , beta);
```

```
if(time ] 60 seconds) break;
```

```
++i;
```

{ 81 % : 使用迭代加深 , 在进行第 d 层搜索时往往可以从第 d-1 层的搜索结果中获取一些启发信息 , } { 69 % : 往往可使整个搜索过程比不比迭代加深的搜索更快。 }

2.注册棋盘点击事件

当玩家点击棋盘时 , 在相应的位置上添加棋子 , 然后电脑下子 , 电脑下子时屏蔽棋盘点击。

3.2.6蓝牙对弈设计

{ 49 % : 蓝牙对弈需要分平台实现数据传输的功能。 } { 44 % : 因为客户端和服务端都是手机 , 因此ios采用的是GameKit , android采用的是bluetooth来作为蓝牙开发。 } Android没有用BLE是因为BLE需要设备为4.3及4.3设备上 , 本游戏为了适配更多的手机 , 所以采用了bluetooth。 然后分别实现搜索设备、连接设备、发送数据、接受数据、断开设备等接口。

3.2.7 wifi对弈设计

wifi对弈需要需分平台开发。 { 40 % : Android 和 ios端要统一实现搜索设备、连接设备、发送数据、接收数据等接口。 } { 100 % : 在Android中对Wifi操作 , android本身提供了一些有用的包 , 在android.net.wifi包下面。 } ios端采用Socket进行通讯。

3.2.8 结束页设计

结束页因为需要显示赢棋方、时间和步数 , 因此当游戏结束时应该把这些参数传到游戏结束页中。 结束还要提供悔棋 , 重玩 , 返回主页等操作 , 这些操作可由外面注册回调函数。 为了美观 , 可使结束也呈半透明状态 , 因此不应该把结束页做成一个新场景 , 可以采用layer。

4 系统实现

4.1 工程搭建

4.1.1 环境搭建

1. cocos环境安装 :

在官网<http://www.cocos.com/download/#下载cocos2d-x> , 然后执行cocos2d-x目录下的setup.py即可安装。

2. android开发环境安装 :

需要的工具有adt, ndk, sdk, ant。 编辑工具为android-studio; 将这些工具下载后, 配置相应环境变量即可。 Mac下的环境变量配置在用户根目录下的.bash_profile文件中。 环境变量配置总体如下:

```
# Add environment variable COCOS_CONSOLE_ROOT for cocos2d-x

export

COCOS_CONSOLE_ROOT=/Users/liubo/cocos2d-x-3.8.1/tools/cocos2d-console/bin

export PATH=$COCOS_CONSOLE_ROOT: $PATH

# Add environment variable COCOS_TEMPLATES_ROOT for cocos2d-x

export COCOS_TEMPLATES_ROOT=/Users/liubo/cocos2d-x-3.8.1/templates

export PATH=$COCOS_TEMPLATES_ROOT: $PATH

# Add environment variable ANDROID_SDK_ROOT for cocos2d-x

export ANDROID_SDK_ROOT=/Users/liubo/Android_tools/adt-bundle-mac-x86_64-20131030/sdk

export PATH=$ANDROID_SDK_ROOT: $PATH

export PATH=$ANDROID_SDK_ROOT/tools: $ANDROID_SDK_ROOT/platform-tools: $PATH

# Add environment variable NDK_ROOT for cocos2d-x

export NDK_ROOT=/Users/liubo/Android_tools/android-ndk-r10c

export ANDROID_NDK_ROOT=$NDK_ROOT

export PATH=$NDK_ROOT: $PATH

# Add environment variable ANT_ROOT for cocos2d-x

export ANT_ROOT=/Users/liubo/Android_tools/apache-ant-1.9.4/bin

export PATH=$ANT_ROOT: $PATH

#Add ANDROID_HOME

export ANDROID_HOME=/Users/liubo/Android_tools/adt-bundle-mac-x86_64-20131030/sdk
```

4.1.2 创建工程

此处用到了cocos创建项目命令：`cocos new programName l lua p packageName d path`；其中new后面是工程名，-l后面是语言（可选项为lua、c++、js），-p后面是工程的包名。此处我创建的是五子棋项目，项目名称为WuZiQi，使用的是lua语言。然后进入WuZiQi/frameworks/runtime-src/proj.ios_mac目录下，点击WuZiQi.xcodeproj即可在xcode中打开该工程。

工程结构图如下：

图4.1 工程结构图

external为我封装的外库，提供一些蓝牙和wifi的操作。Classes下面的AppDelegate为程序的入口。Resources为下面的res为资源文件，src下为lua代码，实现了游戏相关的逻辑与界面。

4.2 游戏首页实现

游戏首页分为五个按钮和一张背景图。 { 40 %：每个按钮都加入了对应的点击事件，点击对弈模式等按钮时进入相应的游戏主场景。 } 点击游戏规则时会进入讲解游戏规则的场景，为五子棋玩家讲解一些五子棋术语。游戏首页如图4.2。

图4.2 游戏首页

4.3 游戏规则界面实现

游戏规则讲解场景如图4.3，左右滑动可切换游戏规则的讲解。点击跳过时可终端游戏的讲解，回到游戏首页。

图4.3 游戏规则讲解场景

4.4 游戏主界面实现

游戏主页提供了声音、重玩、悔棋、提示、求助、时间显示、棋盘等空间。如图4.4点击悔棋时会撤掉刚下的棋子，标注当前下子的小红点也会撤回到上一步棋子上。 { 43 %：点击重玩是会清空棋盘上的棋子，时间清零。 } 点击提示时会调用AI算法，在当前最优的位置下子。点击声音是会出现下拉菜单，展示出声音和音效。暗色代表关闭状态，亮色代表开启状态。如图4.5

{ 47 %：图4.4 游戏主界面 图4.5 展示音效音乐图 }

4.5 游戏结束界面实现

当五子连珠或者和棋时，弹出结束界面。显示赢棋方或者和棋，并且显示了游戏的时间，赢棋方所用的步数，还可以重新挑战、返回主页、悔棋等。如图

图4.6 游戏结束页

4.4 双人对弈实现

由于棋盘类已经分装好了下子和切换下子方的功能，因此双人对战只需要注册点击棋盘的回调函数即可，当点击棋盘时，执行回调函数。

4.5 人机对弈实现

人机轮流下子。 当电脑下子时，显示电脑正在下子状态，如图4.7。

图4.6 人机对弈场景图

4.6 蓝牙对弈实现

当点击蓝牙对战时，如果是ios端，则出现搜索蓝牙状态，如图4.8。 如果是android端时，出现授权、创建游戏和加入游戏的选项，如图4.9。 当点击悔棋时，如果当前已下子方不是自己，则提示不能悔棋，如图4.10，否则发出悔棋请求，如图4.11。 对方收到请求如图4.12，对方同意请求或者拒接请求时己方会收到对应的消息，如图4.13。 重玩操作类似，此处省略。 当点击提示时，弹出提示不可用信息，如图4.14。

图4.8 ios蓝牙搜索图 图4.9 android蓝牙搜索图 图4.10 悔棋提示图

图4.11 发送悔棋请求图 图4.12 接受悔棋请求图 图4.13 悔棋请求应答图

图4.14 提示功能不可用的友好提示图

4.7 wifi对弈实现

还没有实现，正在开发中

结 论

经过2个多月的开发，本款五子棋游戏终于独立开发完成了。 游戏中包含双人对弈，人机对弈，蓝牙对弈以及wifi对弈等多种玩法，但还有很多不足。 比如人机对弈中AI的棋力不是很高，蓝牙对弈中偶尔会出现数据单线传输的情况。 但在以后的日子里我会继续开发，使这款游戏得以完善。 通过本次开发，我更加熟练的掌握了cocos引擎，了解和掌握了一些博弈算法，ios和android的蓝牙开发以及跨平台的封装。 我知道这还远远不够，知识是无止境的，我所了解和掌握的只是冰山一角。 在以后的路上，我要更加深入的理解编程思想，多去了解ios和android等平台应用的开发技术。

致 谢

首先我要感谢王强老师。 本篇论文是在王老师的帮助下完成的，王老师是个认真负责的人，给予了我很多学术和技术上的指导。 在这里我感谢王老师的帮助，感谢您教会我这么多，您教会我的不只是技术，更是一丝不苟的精神。

其次我要感谢大学这四年来老师们对我的培养，感谢学校给予了我们一个这么温馨宁静的学习环境。 学校的

3+1制度让我得到了很快的成长。 让我了解了社会形式，更加清楚的知道了自己的发展方向。

刚开始选择游戏开发的原因很简单，因为工资比较高，又可以锻炼逻辑思维。 随着技术的深入，我发现游戏开发所吸引我的不只是薪资的高低，还有它的多变性。 因为每一款游戏都不一样，算法各异，技术不同。 我需要不断的去学习，在有限的时间内解决未知的问题。 我觉得很有挑战性，虽然过程很令人苦恼，但结果确让我增加了一份成就感。 我所收获的不只是经验，还有技术上的提升。 不过这一切归根结底还是要感谢学校，给予了我们一个很好的平台，让我们能够越走越远！

参考文献

[1] 陈其.PC游戏编程-人机博弈[Z]. [http: //vdisk.weibo.com/s/dt-T5jclT_KtT](http://vdisk.weibo.com/s/dt-T5jclT_KtT) , 2005.

[2]Roberto Ierusalimschy. Programming in Lua[Z]. [http: //vdisk.weibo.com/s/z63ACRNxK4cLS](http://vdisk.weibo.com/s/z63ACRNxK4cLS) , 2005-07-26.

[3] 肖文吉. Cocos2d-x 3.x游戏开发实战[M].北京： 电子工业出版社，2015-01： 35-260.

[4] 李刚.疯狂ios讲义上[M].北京： 电子工业出版社，2015-04: 24-254.

[5] 王刚，杨巨峰.C++ Primer[M].北京： 电子工业出版社，2013-09: 29-273.

[6] 王菁，赵元庆.UML建模设计与分析[M].北京： 清华大学出版社，2013-07: 24-109

[7] android.bluetooth[Z]. https://developer.apple.com/library/mac/documentation/GameKit/Reference/GameKit_Collection/.

[8]apple.GameKit.framework[Z]. https://developer.apple.com/library/mac/documentation/GameKit/Reference/GameKit_Collection/.