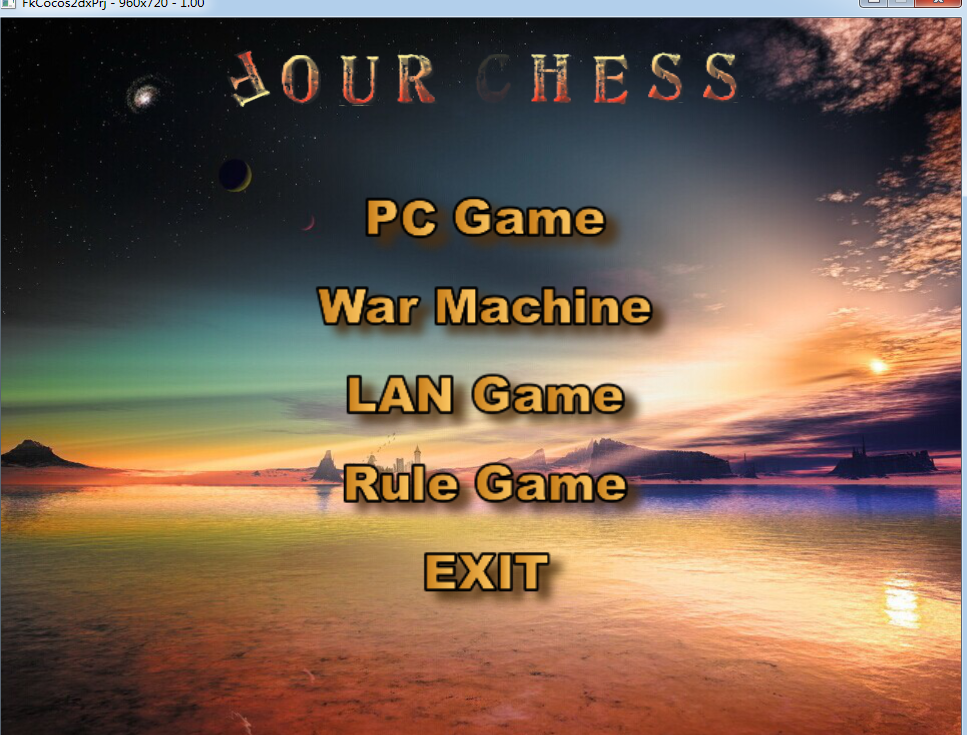
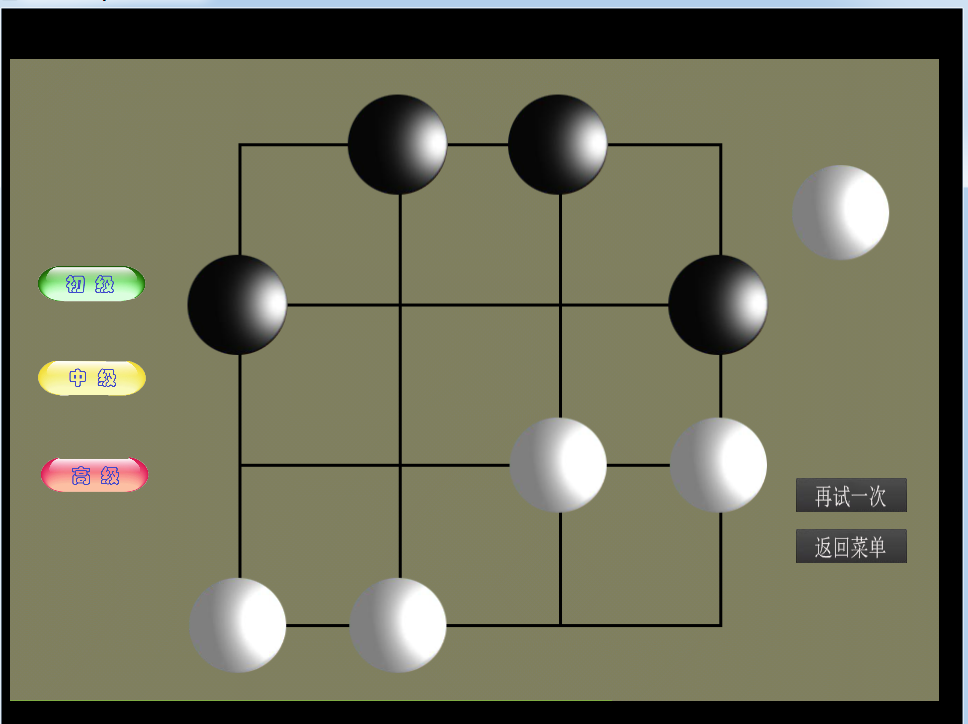
**个人简历**

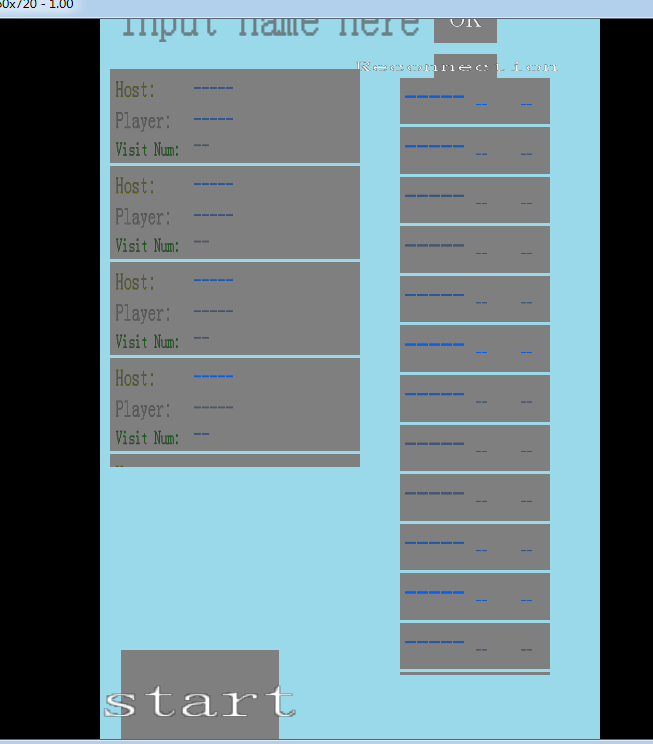
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **姓 名** |  | | **性 别** |  | **出生年月** |  | **政治面貌** |  |
| **毕业院校** |  | | | | **学历** |  | **专业** |  |
| **联系电话** |  | | | | **Emai** |  | **通讯地址** |  |
| **主要学习和工作经历：** | | | | | | | | |
| **何时何地获得何种奖励或荣誉：** | | | | | | | | |
| **工作经历：**  **2012.09~至今:**     **...科技有限公司    软件工程师**  **保卫农场游戏**  **....** | | | | | | | | |
| **择业目标** | | **C/C++、cocos2d-x 、WIN 32编程、Unity3D、图形图像处理、** | | | | | | |
| **计算机水平**： | | | | | | | | |
| **专业技能：**    开发语言：C/C++、Objective-c以及汇编语言SQL；    游戏引擎：cocos2d-x、cocosbuider界面开发软件、Unity3D    开发工具：Visual studio 2010、XCode、sql 2008、PLSql等开发工具    内存管理机制：对内存泄露、野指针、引用计数的原理都有深刻的理解    面向对象：封装、继承、多态、重载以及虚函数的原理    Win32SDK编程    Windows API环境下编写多线程，各种线程同步对象的熟练使用；    设计模式： MVC模式、观察者模式、代理模式、单例模式    数据结构：队列和树    数据库：oracle、存储过程    软件工程，OOA，OOP，OOD    数学建模、程序建模    socket网络编程进行网络通讯应用程序的设计和开发的能力；    开发文档：编写项目开发文档（设计文档、需求文档）、设计UML图、用Visio流程图来设计程序 | | | | | | | | |
| **项目经验：**    **2012.09~至今:**     **...公司    软件工程师**  **项目名称：KillGame（手游）**  **开发工具：Visual studio 2010**  **开发语言：C/C++**  **游戏引擎：cocos2d-x**  **项目描述：**   1. 该游戏来自于生活中玩的杀人游戏，描述的是警察、小偷、平民、法官之间的故事，本款软件充当了法官的角色。   **项目分工：**项目组合作完成，审查文档，游戏规划、网络通讯模块、角色管理、各个角色、信息显示等等、界面设计，我只担任其中的…模块  **开发技能：**   1. 需求、规划（对象图、类图、UML图）、流程图等文档进行设计。（见附录） 2. 各场景之间的特效切换 3. 用CCLayer建立新的图层，将图片放在图层上面 4. 用CCMenu添加多个控件 5. 用setPosition()设置每张图片的位置 6. 利用setScale()将精灵放大与缩小 7. 利用setRotation()改变精灵的视角 8. 用三角函数制作椭圆型的玩家会谈 9. 用ccTouchBegan、ccTouchMoved来完图片的触摸和拖动 10. 通过setTouchEnabled来屏蔽点击事件 11. 多个触摸事件之间的优先级 12. 用getLocation获得当前点击坐标 13. 利用socket实现局域网中数据的传输，将每一个玩家的信息发送给房主，房主充当法官的角色，将计算好的信息传回每一个玩家。 14. 利用随机数，随机每个玩家的角色。 15. 利用cocostudio做界面设计，并导入cocos2d中使用。 16. 设置层与层之间的添加、覆盖关系，以及层与层之间，确定子类与父类之间的划分 17. 利用initWithFile更改精灵图片。 18. 用文件保存用户信息。 19. 用CCProgressTimer(进度条)和字符串拼接，显示用户等待所需要的时间 20. 用scheduleOnce()定时器与gettimeofdayCocos2d()调用系统时间进行发言倒计时。 21. 搭建安卓环境，将游戏导成APK放到手机中使用。 22. 在程序执行过程中，前台继续运行，后台用多线程继续接收消息，互不影响。 23. 屏幕自适应。 24. 利用select检测是否有连接，实现一对多的连接。   **项目心得：1、**熟练了cocos2d-x引擎的使用,发现想要写好一个程序不是光靠网页就行了,有些时候只能通过源码,看底层的实现。  2、经过这次编程我发现了文档的重要性,以前写的都是一些只有几百行的代码,一会就写好了,不需要什么设计,需求,都不需要,只要会一点代码就能弄出来的,在做这个时程序时我发现要没有文档的话一天话费在交流上面的时间都去了过半,项目根本没办法进行下去。  3、我们的分工也不完整,没有按照面向对象的思路来分工,还是按照面向过程的思路在思考,导致我们的在写项目的过程有些人忙的要死,有些人却闲的没事.在这个项目中我体会到了面向对象的重要性，在大项目中，如果用面向的思路，能大大的提高团队的编程效率和合作的效率，不会出现一人做事，多人围观的状态。      **四子棋（手游）**  **开发工具：Visual studio 2010（c++）**  **开发语言： c++**  **开发引擎：cocos2d-X**  **项目描述：**   1. 四子棋小游戏，实现非联网状态下两人对弈、联网状态下两人对弈、人机对弈三个模块。 2. 简单规则：两颗棋子吃掉一颗棋子   **项目分工：**需求、规划、设计与代码实现，  **开发技能：**   1. 需求、规划（对象图、类图、UML图）、流程图等文档进行设计。（见附录） 2. 用touchesEnded响应棋子的点击事件 3. 用getLocation获取当前坐标 4. 将棋子添加到CCSprite里面 5. 用for循环将棋子添加到棋盘上 6. 分别用CGContextSetLineWidth、CGContextSetStrokeColorWithColor、CGContextMoveToPoint、CGContextAddLineToPoint来设置棋盘线的宽度、颜色、起点、终点 7. 通过改变棋子的中心坐标(pView.center)来移动棋子 8. 游戏结束时，用UIAlertView弹出提示框提示用户赢棋 9. 利用MVC模式，将判断是否能吃子、赢棋、走棋、放在后台，将棋子的移动图像放在前台 10. 前台代码用cocos2d-x写，后台代码用c++写。为的是能跨平台 11. 用二维数组存放棋盘上棋子的情况 12. 用算法优化吃子计算 13. 将坐标转换成为二维数组的下标 14. 将棋谱用二维数组保存下来，优化赢棋 15. 通过算法实现AI（人机对战） 16. 利用socket实现数据传输 17. 利用队列存放需要向另一方发送的信息，解决数据的丢失和网络延时的问题。 18. 在程序执行过程中，前台继续运行，后台用多线程继续接收消息，互不影响。 19. 搭建安卓环境，将游戏导成APK放到手机中使用。   **项目心得：**通过视图和逻辑的分离设计，深刻地理解了MVC设计模式    **象棋（手游）**  **开发工具：Visual studio 2010（c++）**  **开发语言： c++**  **开发引擎：cocos2d-X**  **项目描述：**   1. 象棋小游戏，实现非联网状态下两人对弈。   **项目分工：**需求、规划、设计与代码实现，  **开发技能：**   1. 需求、规划（对象图、类图、UML图）、流程图等文档进行设计。（见附录） 2. 用touchesEnded响应棋子、棋盘的点击事件 3. 用getLocation获取当前坐标 4. 将棋子添加到CCSprite里面 5. 用for循环将棋子添加到棋盘上 6. 通过改变棋子的中心坐标来移动棋子 7. 游戏结束时，用UIAlertView弹出提示框提示用户赢棋 8. 利用MVC模式，将判断是否能吃子、赢棋、走棋、放在后台，将棋子的移动图像放在前台 9. 前台代码用cocos2d-x写，后台代码用c++写。为的是能跨平台 10. 用二维数组存放棋盘上棋子的情况 11. 用算法优化吃子计算 12. 将坐标转换成为二维数组的下标 13. 将棋谱用二维数组保存下来，优化赢棋 14. 棋子自动寻径算法 15. 搭建安卓环境，将游戏导成APK放到手机中使用。   **五子棋（手游）**  **开发工具：Visual studio 2010**  **开发语言：C#**  **游戏引擎：Unity3D**  **项目描述：**  1、五子棋的小游戏，实现两人对弈和人机对战AI  **项目分工：**紧担任五子棋项目的设计以及所有功能实现  **开发技能：**   1. 用GUI绘制控件 2. 用PlayerPrefs类保存、读取数据 3. 用Application类实现场景切换和程序退出 4. 用二维数组记录棋盘上的棋子情况 5. 用GameObject类型的数组aobjChess保存黑棋、白棋的对象 6. 给棋盘添加碰撞体，在OnMouseUp()中获取用Input.mousePosition获取鼠标（触摸）的坐标 7. 将坐标装换成二维数组的下标 8. 用Instantiate克隆aobjChess数组中黑棋或白棋对象 9. 判断落棋、赢棋 10. 游戏结束时，提示用户赢棋。 11. 用户点击 “再玩一次”按钮，用Application.LoadLevel()切换本场景。 12. 优化坐标下标的转换，以及赢棋算法 13. 优化AI算法 14. 搭建安卓环境，将游戏导成APK放到手机中使用。   **项目心得：**熟悉Unity3D引擎，并用其完成项目。  **密码锁（u3d版手游）**  **开发工具：Visual studio 2010**  **开发语言：C#**  **游戏引擎：Unity3D**  **项目描述：**  1、设置密码  **项目分工：**密码锁的设计以及所有功能实现  **开发技能：（匀减速、振荡、归中）**   1. 拖拽图片到Scene中，创建对象，在Inspector中设置对象的位置以及旋转角度 2. 鼠标点下OnMouseDown时，用Input.mousePosition获取的鼠标坐标转换成相对于密码锁圆点的坐标 3. 鼠标拖拽OnMouseDrag时，密码锁圆盘随着鼠标转动——已密码锁圆心为原点，计算出鼠标当前位置相对于坐标系的角度，与上次角度的差为当前要旋转的角度，用transform.Rotate()实现旋转 4. 鼠标抬起OnMouseUp时,通过当前的速度进行匀减速运算 5. 振荡归中算法 6. 用Texture[]保存图片，将密码锁圆盘旋转的角度转换成要显示图片的对应Texture[]的下标 7. GameObject.Find()获取游戏对象，用GetComponent<>可以获取脚本 8. 用定时器实现长按 9. 搭建安卓环境，将游戏导成APK放到手机中使用。   **项目心得：**熟悉Unity3D引擎，并用其完成项目。  **密码锁cocos2d版（cocos2d版手游）**  **开发工具：Visual studio 2010（c++）**  **开发语言：c++**  **开发引擎：cocos2d-X**  **项目描述：**  1、 玩家通过旋转屏幕上的转盘改变数字，当转到正确数字时可以解锁。  **开发技能：**   1. 用touchesEnded、touchesBegan、touchesMoved响应转盘的点击、旋转事件 2. 将数字都贴到圆盘上面，用setRotation旋转圆盘 3. 利用aTan()计算需要旋转的角度，并矫正。 4. 通过角度计算出最下面的数字，并通过initWithFile更换数字所对应的图片。 5. 利用重力加速度的公式计算模拟单摆动作，当停止拖动时能因为重力的原因不能马上停止旋转，让密码锁更接近现实。 6. 利用算法让数字自动归中，停在锁钩的正中间。 7. 定时器记录时间，实现长按修改改密码、短按解锁的功能。 8. 搭建安卓环境，将游戏导成APK放到手机中使用。   **责任描述：**只负责人物的翻转和人物移动模块  **多线程买票（控制台项目）**  **开发工具：Visual studio 2010（c++）**  **开发语言：c++**  **开发引擎：cocos2d-x**  **项目描述：**  1、 利用多线程实现多个售票员同时卖票  2、 通过使用同步对象实现不重票、漏票、跳票等等  **开发技能：**  1、  多线程  a)   用CreateThread()创建线程  b)   分别用SuspendThread、ResumeThread、TerminateThread来挂起、恢复、关闭线程  c)   用WaitForMultipleObjects等待线程结束  d)   封装线程  e)   卖票程序继承自封装好的线程  2、  用汇编语句编写卖票号增加的语句  3、  同步对象  a)   分别用InitializeCriticalSection来初始化临界区  b)   用EnterCriticalSection进入临界区  c)   分别用LeaveCriticalSection来离开各个临界区  d)   通过DeleteCriticalSection关闭临界区  4、  封装同步对象隐藏用户对SDK的操作，比如（什么时候开始、结束同步对象）  5、  调用封装好的同步对象，实现不跳票、不卖重票等等  **责任描述：**紧担任多线程模块 | | | | | | | | |
| **自我评价：** | | | | | | | | |
| **大学所学课程：** | | | | | | | | |

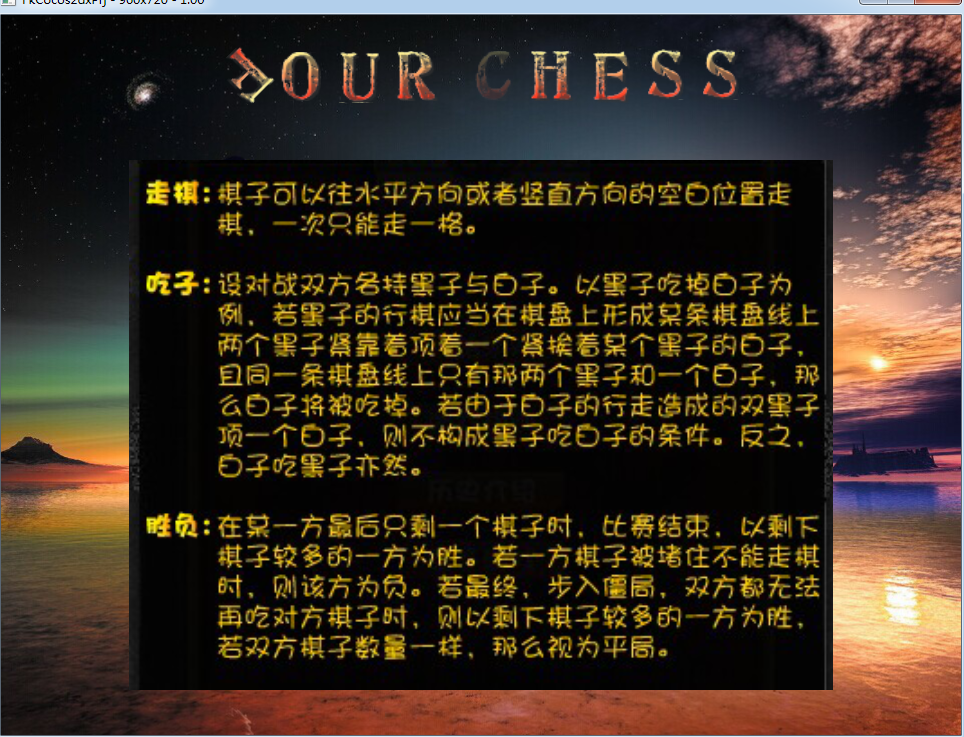
各项目截图:

1. 四子棋





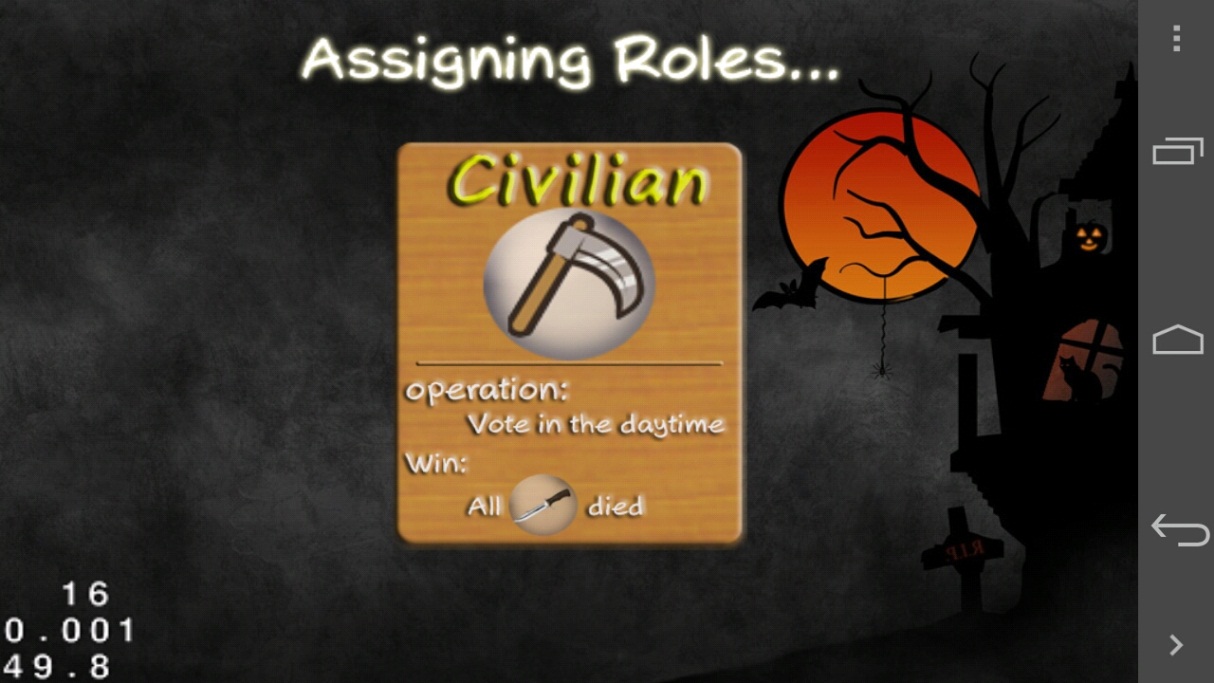




1. KillGame

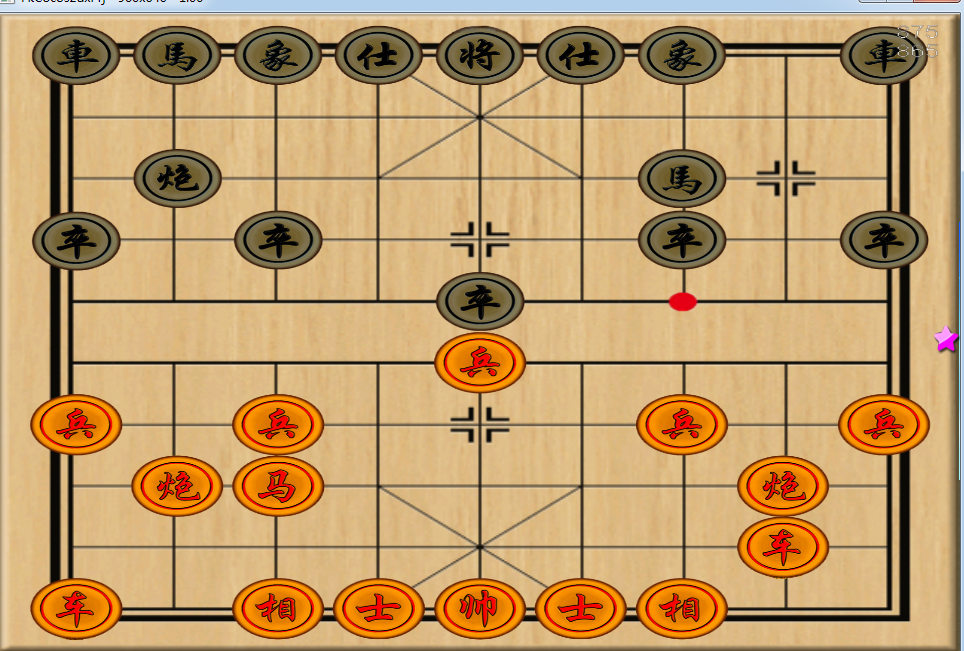






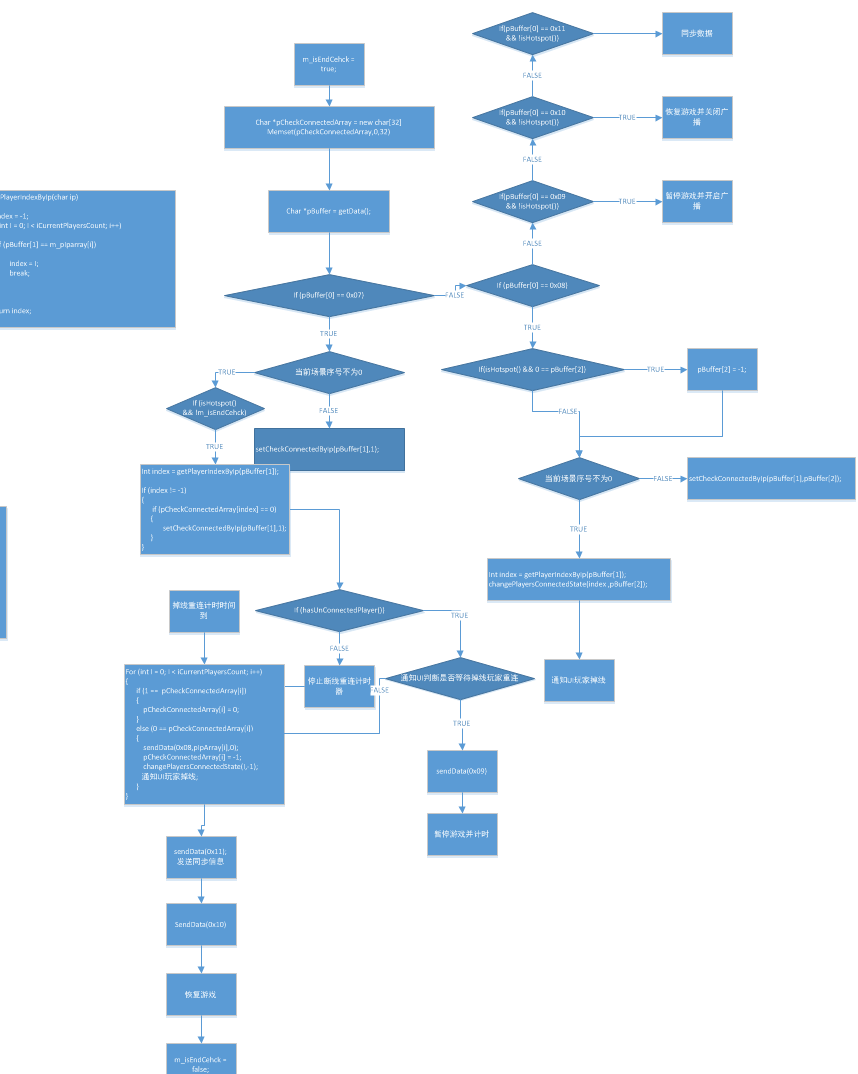
1. 发
2. 象棋







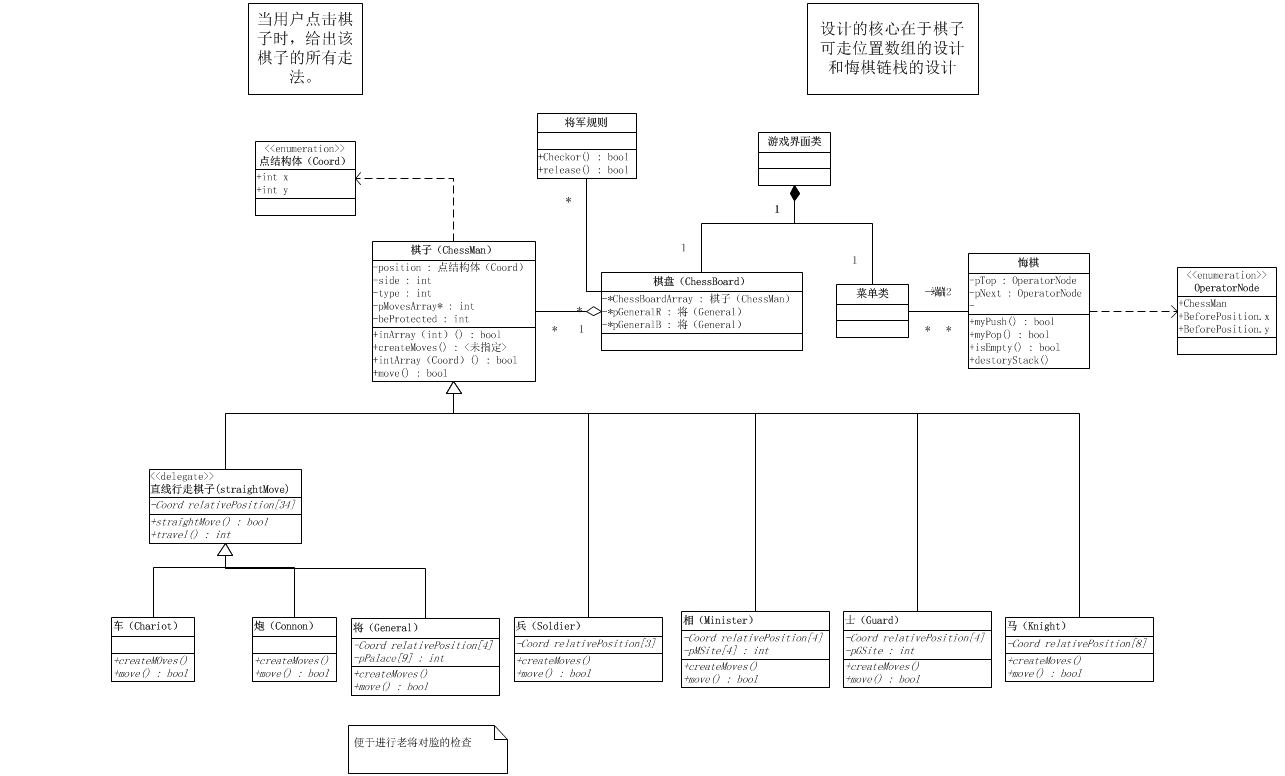
附录1…KillGame部分流程图



附录2…四子棋部分规划图  判断是否可以吃子



**附录3    象棋UML图**



**附录4   象棋部分流程图**

