五子棋游戏设计文档

前言

此为 Qt 设计游戏二人对战五子棋的设计文档，本文档分为三个部分：

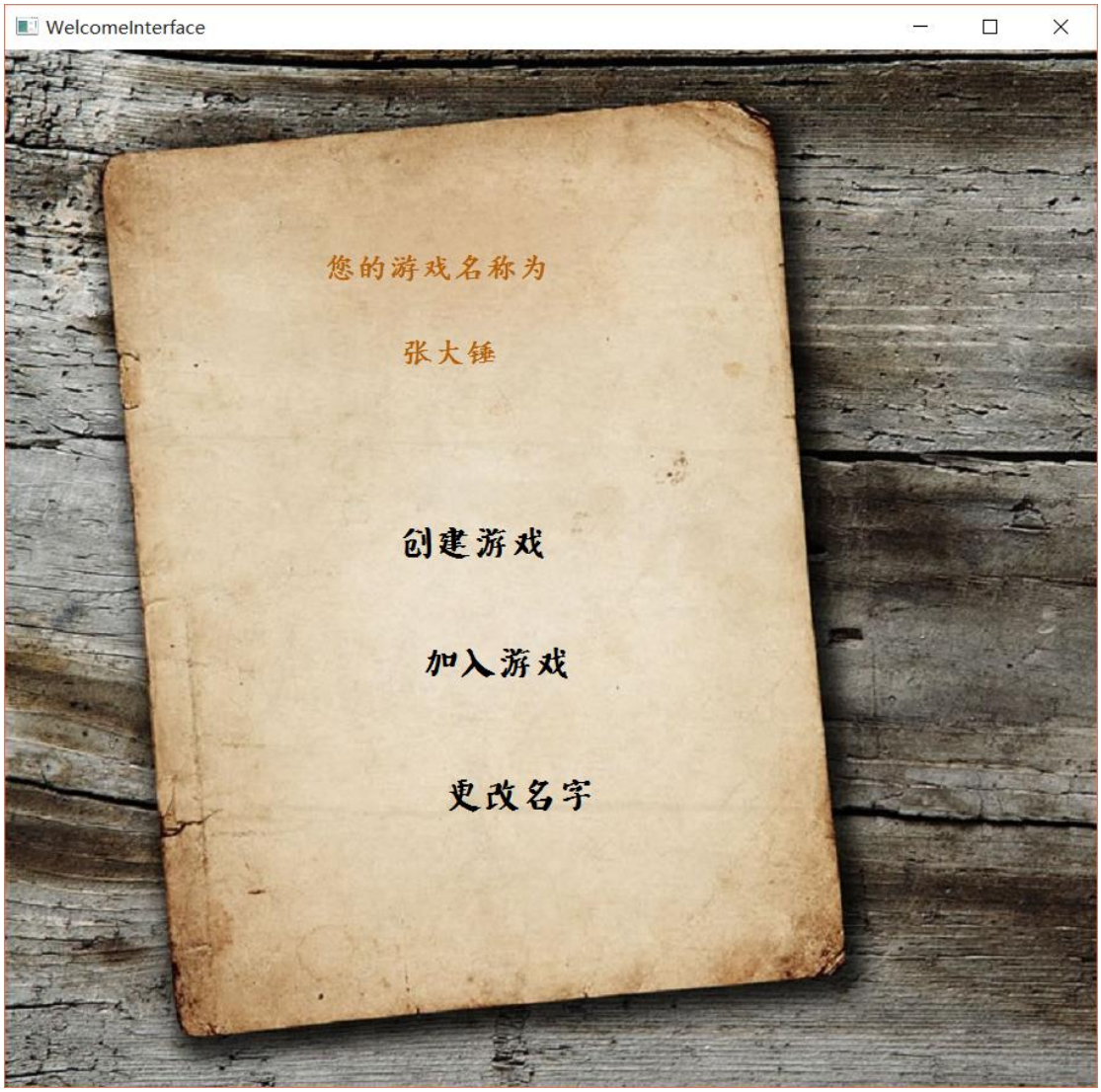
1. 游戏特色介绍
2. 游戏设计说明
3. 游戏设计过程中遇到的问题及解决方法。

正文

特色介绍

本游戏为使用 TCP 连接的二人对战五子棋游戏，一方玩家点击“创建游戏”新建游戏，另一方玩家点击“加入游戏”根据提示的 IP 地址连接游戏即可对战。

代码可在 https://github.com/ytl13508111107/chess 下载



连接游戏，当双方玩家都准备后游戏开始，每回合有 20 秒的思考时间，超过时间则系统自动下子。

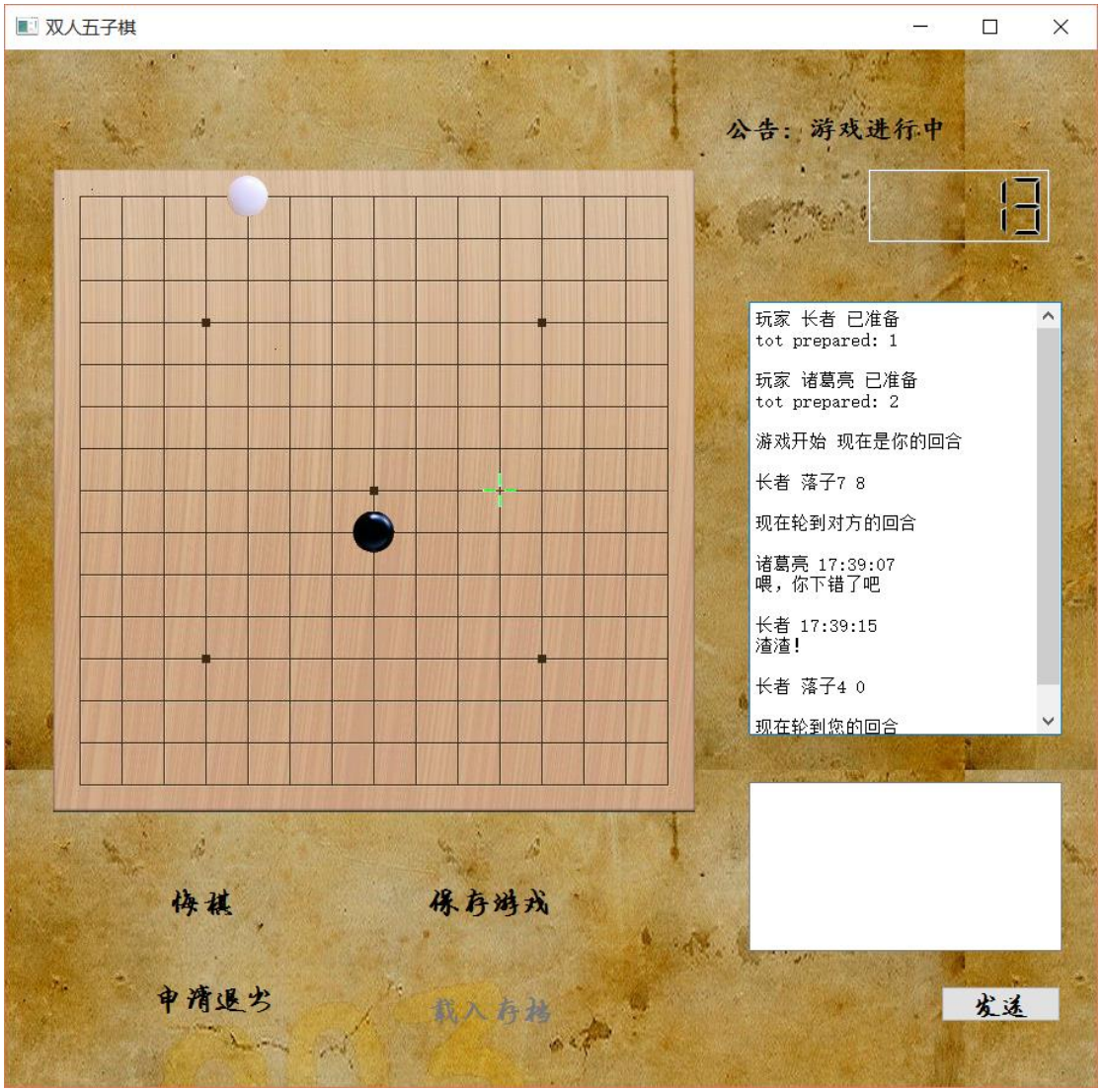
系统会自动记录玩家的每一次落子事件并保存为本地备份文件，以供在游戏突然退出时将来能够恢复局面。

游戏过程中可以随时保持棋局。并在下一次游戏开始前载入存档。

游戏过程双方可以自由发消息，并可申请悔棋或退出，均需对方同意后才能进行。

如对手非正常退出超过 5 秒无响应，或传输过程中发生网络错误，均视为连接异常，系统会自动保存当前局面并提示消息退出。

当一方完成五连珠，系统弹出提示宣告游戏结束



游戏设计说明

游戏被分为信息保存，界面交互，网络通讯三个主要模块，分别对应三个类 *GameInfo*、*GameInterface*、*Top*

1.网络通讯

*Tcp*类的主要工作为服务端，客户端的创建，网络连接的建立，对待发送的信息进行打包并发送，接收对方传来的消息，并组织成包，检查包的完整性，以及确保游戏过程中的连接状态等。

当点击创建游戏时通过*Tcp*创建*Server*，并监听对应端口，期间若玩家取消则关闭监听。

当点击加入游戏时*Tcp*创建*Client*并视图和对应*IP*端口的*Server*建立连接，连接成功发送成功信号交给上层处理。同时作为*Server*的一方传递游戏初始信息给*Client*。

*Tcp*之间的数据通信模拟了网络层和传输层的传输过程：

对于一段待发送的包含多个字符串的游戏信息，先用’|*n*’将之层层打包为一个字符串，再在字符串前加上指令头以说明信息的含义。对于文本信息还需要将文中的空格和回车替换掉以免读取的时候被漏掉。之后给字符串加上特定的信息头和尾组织成包，将整理好的包发给对方主机

接收方收到信息后先检查是否是信息头，收到信息头后进入接收状态等待对方传递完整信息，将之后收到的所有信息压入缓存，直到收到信息尾。收到信息尾后结束接收状态。取出之前的所有缓存组织成包，再层层解压为指令和游戏信息交给上层处理。

接收时如果收到信息头后*5*秒仍未能收到信息尾，被视为在传输过程中网络连接出现问题，发送断开信号交给上层处理。

同时，自双方连接建立连接后，双方的*Tcp*每隔*5*秒都会给对方发送一个“确认心跳”信号，对方收到信号之后必须立即回复“我还活着”，如果超过*5*秒没有回复，被视为对方非正常下线，发送游戏连接断开信号

交由上层处理。

2.信息保存

信息保存分为用户信息和游戏信息，

用户信息包括id，玩家名称，头像/待添加/等可设置的属性，会在第一次建立连接时传给对方。

游戏信息为游戏进行时产生的所有资料。包括当前落子方，自己先手还是后手，每个格的棋子状态，聊天信息，思考时间，开始时间等等。当游戏过程中游戏信息发送变化时，传递对应改变信号给对方主机。

3.

界面部分负责游戏界面的绘制，时刻检查并调整游戏状态（开始，结束，准备），根据玩家的鼠标键盘操作相应对应的事件，并修改游戏信息，处理Tcp收到的信息以及自适当的时候通过Tcp传递必要的信号给对手。

包含以下等功能：

1.定时在后台备份玩家的局面信息保存为本地文档，以便在遇到不可预料的情况后玩家能通过备份找回原来的局面。

2.在按下保存游戏时弹出对话框，并检查文件路径合法，保存当前局面信息到路径。

3.在读取存档时打开相应文件，检查合法并载入游戏。

4.在玩家点击鼠标时检查位置的合法性，确认无误后落子，更新游戏

状态并发送对应的 tcp 信息

5.在聊天发送钮被按下时刷新聊天对话框并发送聊天信息给对方。

6.在反悔或申请退出时暂停游戏，并等待对方的回复，若对方超过 5 秒没有回复。视为对方同意申请。

……

遇到的问题和解决办法

一开始的构思是制作一个服务器客户端一体，同时支持多客户端多服务器工作模式的游戏，用户点击创建游戏建立客户端创建房间并进入游戏，其他的用户可以点击加入游戏，程序会自动通过 UDP 获得此局域网上的服务器列表并显示，也可以手工输入 IP 加入游戏。在游戏中可以支持多人的加入，当有两人准备时开始游戏，其他人进入观战模式，可以通过聊天给建议。

本来按照已经完成了游戏和网路的架构，Udp 和 Tcp 的底层网络传输已经写得差不多了，但我犯了一个致命的错误。为了程序的可维护性，我采用了层层打包，实时同步的办法：对于每一个类和它的子类都写一个压缩和解压的 toString 和 receive 函数，任何一个信息的改变，都会触发游戏改变信号，这时程序会将信息层层打包，直到将整个游戏打包成一个字符串，再发给服务器，服务器收到消息后转发给所有在线用户，用户再解压更新数据。

这样做的好处是提高的程序的可维护性，以后要添加新的功能只需要写对应的类和 toString，receive 函数就可以了。但是我错误的估计了网络的传输速度。将整个游戏都发过去的办法明显信息量太大，严重的阻碍了服务器的处理，导致用户和服务器不能很好的传递信息。而只传递差异的话需要对每一个变量都做检查，写起来过于复杂。结果一直到最后都没有完成，只能做一个简单的，没有太多功能的五子棋游戏。

由此我才意识到网络的传输并不像想象中的那样便捷迅速，应该说其实很慢。网络编程，以及 B/S 架构的网络通信的核心在于通过尽量少的消息来告知对方“我要干什么”“我想要你干什么”，以及合理地处理所有的事件。这正是 http 浏览网页，游戏间通讯的主要思想。