斗地主游戏的设计与实现文档

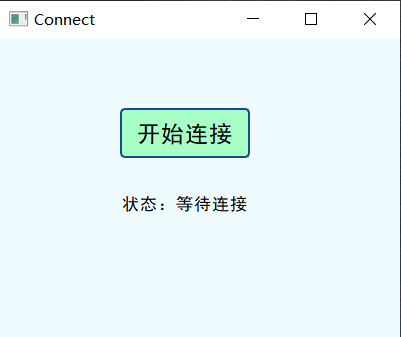
一、总体简介

本程序是对经典游戏“斗地主”的复刻，以在本地运行三个程序的方式，模拟网络通信和网络游戏的编写，并同时练习使用QT进行网络编程。

二、界面及功能介绍

本程序由连接界面和游戏界面两个界面组成，进入游戏后进入连接界面：

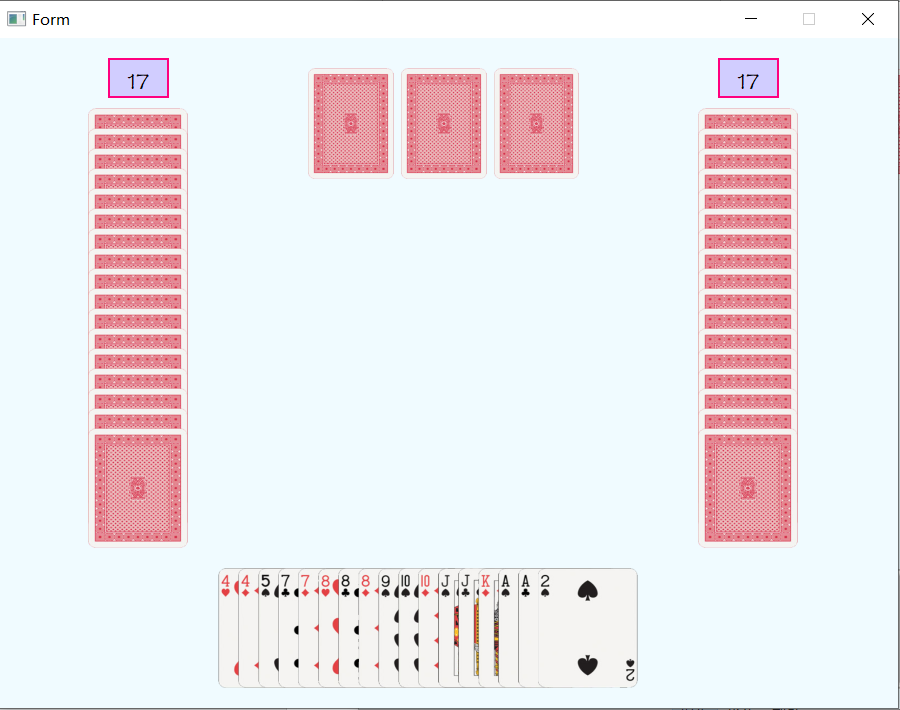
**1.连接界面**



点击开始连接之后，如本地不存在斗地主服务器，则会以这个程序为主机建立斗地主服务器，并监听其他程序的连接请求。如果本地存在服务器，则会作为从机连接到主机（进程）。

**2.游戏界面**

当三个进程相互连接之后，自动跳转到游戏界面。



游戏界面会显示三张底牌、其他玩家手牌数量，你的发牌结果。其中左边为上家，右边为下家。游戏开始时，会随机选择一名玩家叫地主，确定地主的规则为：谁最后叫(抢)地主，地主就是谁的，没人叫就是一开始随机的那个人的。确定地主后，会在所有界面显示底牌，以及身份，并由地主先出牌。

出牌规则包括：单张、一对、三带（三带一/三带二/三不带）、顺子、连对、四带二（四带两单/四带两对）、飞机、炸弹、王炸。

任一玩家出牌结束，则本局游戏结束，同时在所有玩家界面显示游戏结果，并出现重新开始和退出两个按钮。三个玩家都点击再玩一次，则重新开始一局游戏，随机选择地主；选择退出，则三个玩家同时断开连接。

游戏过程截图如下：



三、实现方法

**1.客户端的工作流程及通讯协议**

本程序的客户端工作流程如下如所示：

第一个程序检测到本地没有斗地主进程则会建立通讯进程，而其他的两个进程则唯一的和主机建立通信，如果BC之间想要通信，则需要经过如下途径：

A->B->C

其中A维护两个socket，用于接收BC的请求以及分发消息，BC各自维护唯一的一个socket，用于接收A的要求以及向A汇报自己的信息。其中在A处维护一个server，用于绑定A和BC的socket的读写关系。

通讯协议采用TCP协议，原因是UDP协议不绑定端口，可能会出现丢包的问题，而在本地分配端口比较容易进行，故采用TCP协议。

传输数据的格式方面，采用Json格式进行传输。原因是Json的格式比较自由，而且Qt有现成的高效率的Json库。本程序中设计的Json文档的格式如下：

{

“REQUEST”: int;

“ARGV”: list<int>

}

其中REQUEST代表不同的请求，ARGV表示为了完成这项请求所需要的参数。

**2.三个客户端连接的实现：**

点击开始连接按钮会触发相应的消息响应函数：on\_connectButton\_clicked()，首先socket会先开始尝试连接到"127.0.0.1",8000， 如果连接失败则判定本地不存在斗地主进程，会新建一个server进行对"127.0.0.1",8000进行侦听；连接成功则会将readyRead信号连接到read()槽函数。

服务器在侦听到newConnection()，会通过相应的槽函数，利用nextPendingConnection()生成一个socket来与从机的socket进行通信。

**3.页面转跳的实现**

在MainWindow中维护了一个PlayWidget指针，当有三个客户端连接时，就会隐藏本窗口，并新建一个PlayWidget窗口，同时将socket的槽函数更改为PlayWidget的槽函数，同时为了能够退出，还为PlayWidget添加一个closed()信号，当点击退出时，会相应MainWindow的reshow()函数从而复原主窗口。

**4.叫地主的实现**

进入游戏后，主机使用std::shuffle洗牌，并发送DEAL指令为三个界面发牌。之后，主机会随即决定一名玩家叫地主，并发送CALL指令进行叫地主，程序接收到CALL指令后会激活“叫地主”和“不交”两个按钮。点击任何一个按钮之后会发送CALL\_FINISH指令传回主机，主机收到会发送SHOW\_CALL\_MESSAGE消息令三个客户端显示叫地主提示信息。当三个程序都完成叫地主时，会触发主机的decide\_lanlords()函数来决定地主，并发送BEGIN\_GAME界面来进行初始化游戏界面。

**5.出牌的实现：**

出牌的工作原理大致仿照叫地主。主机通过decide\_play()来决定出牌者，并发送PLAY\_CARDS消息。从机接收到play\_cards()之后，会激活“出牌”“不出”按钮，其中根据是否有牌权来选择性的激活“不出”。

点击“出牌”会检查牌型，牌型检察采用内置类Format来实现。提供两个接口：

bool Format::**isValid**(QList<int>& cards) //检查牌型是否合法

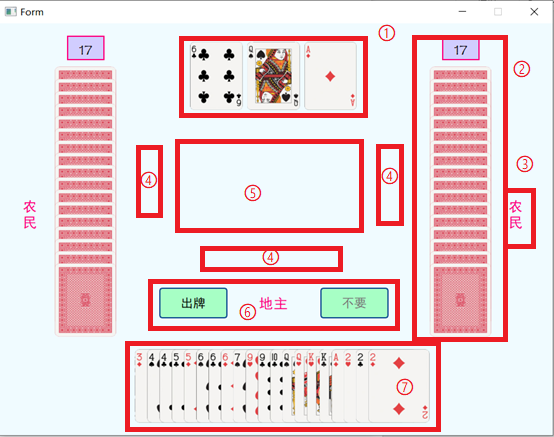
static Format **getFormat**(QList<int>& tcards) //静态接口，用于从出牌中得到牌型

若牌型符合规则，则会判断手牌是否为零，是则向主机发送END\_GAME请求，主机接到后调用end\_game()接口决定各窗口胜负，并显示“重新开始”“退出”两个按钮；若不符合牌型，则会发送PLAY消息，主机收到到会调用decideplay()来决定下一轮的出牌。

同时主机会发送SHOW\_PLAY\_MESSAGE消息，显示出牌信息。

对于 “不出”按钮的处理流程与“出牌”的流程大抵相同。

四、游戏界面介绍



1区为底牌区，采用label控件，setPixmap的方式来显示图片；

2区为其他玩家的手牌区，显示手牌数量，使用label空间的重叠安放实现；

3区为身份信息区，使用label控件显示身份信息。

4区显示游戏提示，消息（不出、叫地主、不叫）。

5区显示其他玩家出的牌，同样是label空间

6去为按钮控件区，提供“叫地主”“不叫”“出牌”“不出”功能，使用StackedWidget空间来进行切换。

7区为手牌区，采用继承自Qlabel的ClickedLabel类来进行显示，重写了MousePressEvent函数来响应鼠标点击事件。