

斗地主 设计文档

2019011336 谢芷钰

1. 实现效果

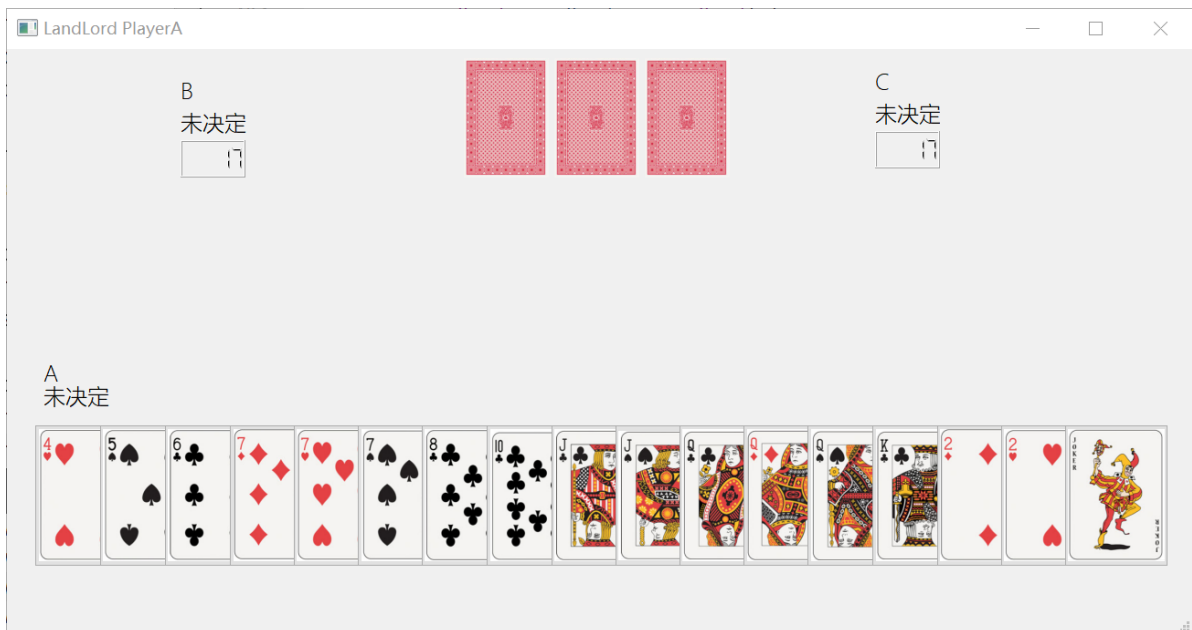
实现了两个界面。运行project，弹出开始界面。

开始界面：



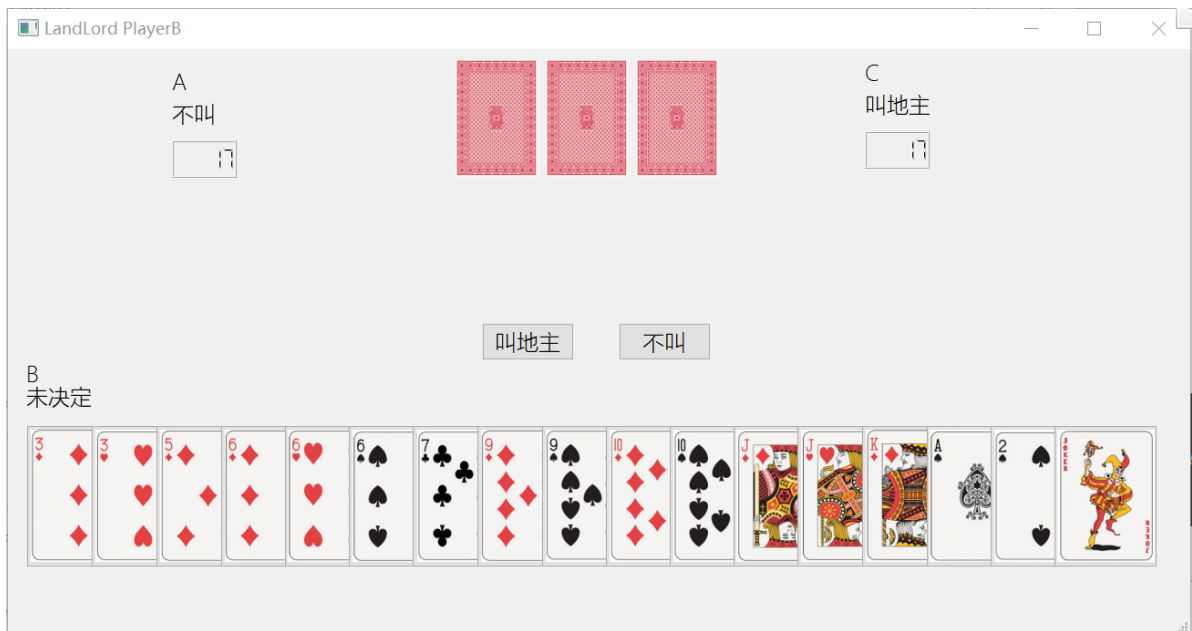
当A,B,C均点击“开始连接”后，程序跳转到游戏主界面。

游戏主界面：



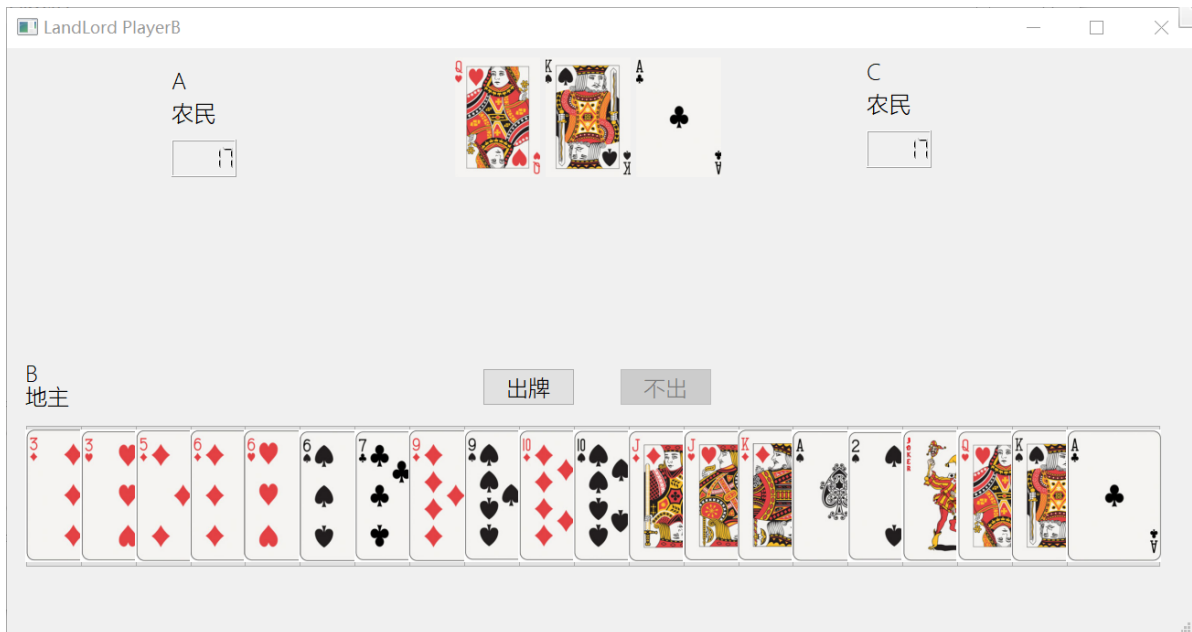
此时随机一个人有先叫地主的权力，之后按照...ABCABC...的顺序轮流叫地主、出牌。

当且仅当轮到自己抢地主时，“叫地主”和“不叫”两个按钮出现在界面上；之前玩家所做的决定都会标在屏幕上。（下图中，C与A已经做完决定，此时轮到B决定）

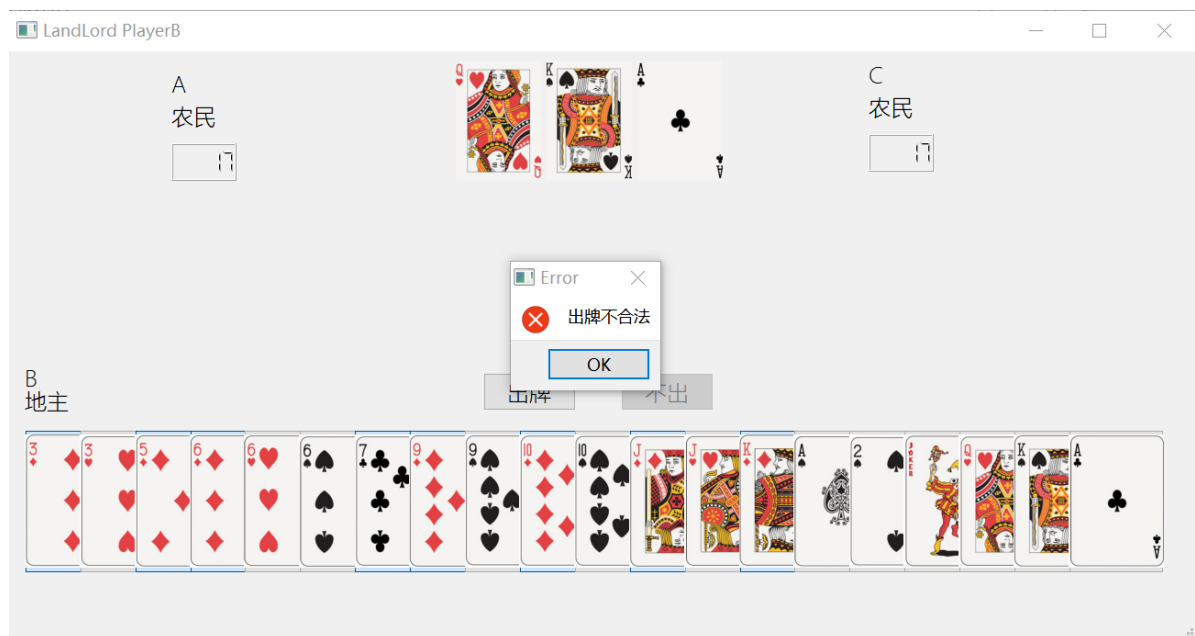


决定完地主与农民后，相应身份显示在屏幕上，三张地主牌亮明，地主牌数计数由17改为20。

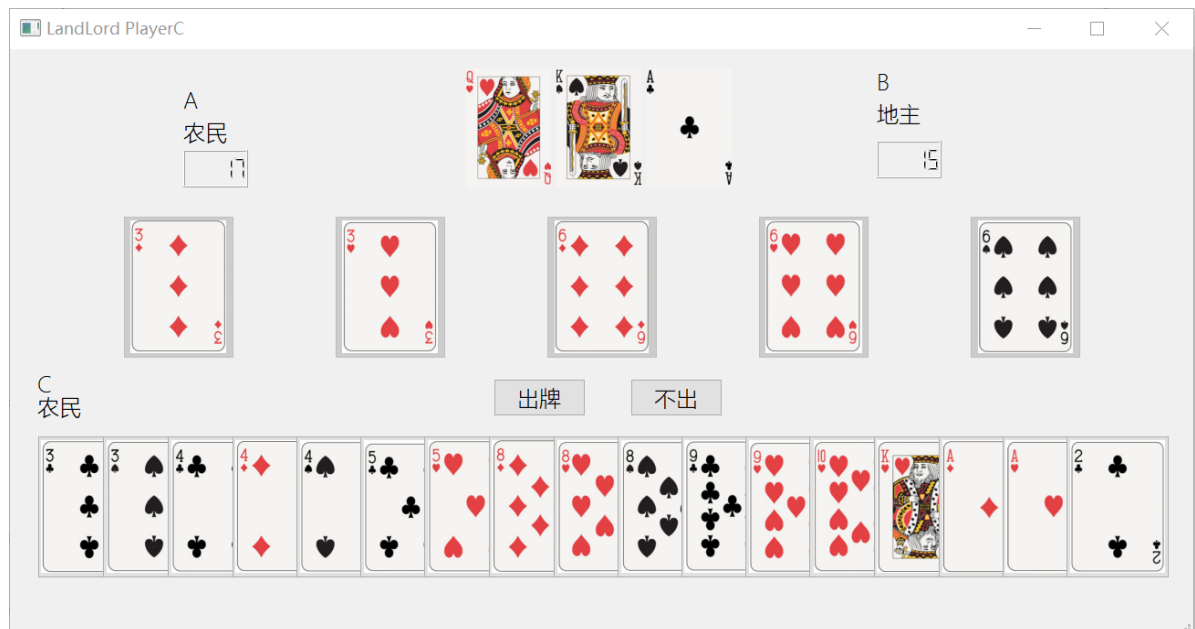
正式开始游戏，由地主先出牌，“出牌”和“不出”两个按钮出现在地主的界面上（不能选择“不出”）。



选择将打出的牌时，牌的边框变为蓝色，若出牌不合法，程序将弹窗提醒。



打出的牌将被显示在界面中间，直到下一组牌被打出为止。打出牌时，界面上剩余的牌数会实时更新。



一人打完牌时，三人的界面均跳出弹窗提示游戏结束并指出自己那方（地主/农民）胜利或失败。

随后，“再玩一局”和“退出”两个按钮显示在界面上。



若点击退出，直接退出游戏，另外两人的界面上弹窗提示“已有玩家退出游戏”；若三人都点击“再玩一局”，则游戏重新开始，回到初始界面，重新随机发牌并随机选择一个人首先叫地主。

2. 设计框架

(这次为A,B,C玩家分别写了一个Qt的project，感觉改bug难度+++)

2.1 主要流程

- 三个程序互相建立连接
- 准备开始游戏，初始化UI界面
- A随机一个先开始叫地主的人，随机每个人的牌和地主牌，把相应信息发送给B和C
- 从随机决定的人开始，按照ABCA的顺序轮流叫地主，决定完把决定发给另外两人，另外两人接收消息之后判断自己是不是应该下一个做决策，如果是就开始叫地主，三人决定后ABC分别判断谁是地主并记录
- 是地主的人自己开始出牌程序，选择出牌或跳过后把相应信息广播给另外两个人
- 如果出牌不合法，程序提示错误，不予出牌，回到选择出牌或跳到阶段。
- 点击“出牌”或“不出”成功后发送消息，另外两个人收到之后更新UI界面（改牌数、显示出的牌或显示不出），判断如果下一个人是自己就开始出牌
- 如果自己出牌后发现牌数为0，或收到别人出牌信息后发现ta牌数变为0，则游戏结束
- 窗口提示游戏胜利或失败，出现“重新开始”或“退出”，点击退出则向另外两人发送消息，接着直接退出程序；点击重新开始也要通知另外两个人。
- 如果自己和其他两个人都选择了重新开始，那么初始化游戏，重新开始。

2.2 网络传输

2.2.1 建立连接

建立连接部分主要实现在 `dialog.h` `dialog.cpp` 中。使用TCP协议。

AB间的信息传输：A作为server，B作为client

AC间的信息传输：A作为server，C作为client

BC间的信息传输：B作为server，C作为client

在点击“开始连接”按钮后，若有server就开始监听，若有client就开始不断尝试连接server；在dialog类内用两个bool变量记录是否已经连接上另外两人，如果均连接上，就告诉mainWindow两个socket的值（A的socket代码中的变量名为 `readwriteSocketA`，其它两个同理），并跳转到mainWindow界面。

2.2.2 消息传输

主要传输的消息有4种类型。分别是

1. 一开始随机决定的牌、第一个叫地主的人、地主牌，在A的 `sendorder()` 和BC的 `recvorder()` 中实现
2. 是否叫地主，在ABC的 `on_beLandlord_clicked()`、`on_notLandlord_clicked()`、`recvLandlordA/B/C()` 中实现。
3. 出牌信息。在ABC的 `sendMessage()` 和 `recvMessageA/B/C()` 中实现
4. 是否重新开始游戏。在ABC的 `on_restartButton_clicked()`、`on_exitButton_clicked()` 还有 `recvMessageA/B/C()` 里特判。

具体传输信息方法如下：

将一副扑克牌编号为0-53，其中小王是52，大王是53，0-51是其余的牌，编号%4对应四种花色，编号/4对应13种牌的大小（编号/4从0-12分别对应3....10JQKA2）。

程序中涉及到扑克牌的内容都采用位运算，2进制下编号对应位为0/1表示对应的牌不在/在组牌中。这样，一组牌的状态可以用 2^{54} 之内的数表示（10进制下为17位），包含花色信息。

初始信息：

A发给B：（17位）B的17张牌的信息+（17位）地主3张牌的信息+（1位）第一个叫地主的人是谁

A发给C：（17位）C的17张牌的信息+（17位）地主3张牌的信息+（1位）第一个叫地主的人是谁

叫/不叫地主信息：

在 `on_beLandlord_clicked()`、`on_notLandlord_clicked()` 中更新UI界面并向另外两个人发送“yes”或“no”字符串。收到时更新UI界面、更新当前判断的地主，判断下一步是否是自己选择，如果是，显示两个按钮。判断是否三个人都已经选择完，如果是，进入 `whoIsLandlord()` 函数，做UI界面准备工作，特别地，地主将三张牌加入自己的牌中，并首先进入 `chooseCard()` 函数

出牌信息：

每个人出牌时，发送给另外两个人如下信息（冗余信息处于方便处理考虑）：

（5位）牌型信息+（17位）打出的牌的具体编号信息+（2位）打出了几张牌

其中，牌型信息用于后面出牌的合法性判断（5位中具体存的信息在2.4.2中解释）；牌的具体编号信息用于在界面中央展示打出的牌；打出牌数用于更新UI界面中的牌数。

重新开始游戏信息：

在 `on_restartButton_clicked()`、`on_exitButton_clicked()` 中分别向另外两个人发送“\$”和“#”表示restart或exit。

2.3 UI界面

主要解释一下牌面的展示。这部分主要写在 `card.h`，`card.cpp` 中。

基于Qt的 `QPushButton` 手写 `Card` 子类，类内存有牌的编号，通过 `id2icon` 函数，将牌的编号对应到例如“S8”的牌面图片文件名，从而设置牌的icon。通过 `setPadding(bool)` 函数，更改牌处于正面或反面状态。

界面中有三个frame，分别用于展示地主牌、上一个人出的牌、自己的牌。

当某个人出牌，后两个frame需要更改时，为了排版美观将原有的所有牌从上一个QHBoxLayout中全部删去，然后新建一个QHBoxLayout将新的牌放进去。

2.4 游戏规则部分

2.4.1 决定地主

在mainWindow中存两个私有成员 firstone 和 landlord。

firstone 是A随机的第一个叫地主的人，landlord 存放谁是地主。

一开始,landlord=firstone，随后三个人做决策时，每次有人叫地主，landlord 就更改成ta。三人做完决策后，landlord 自然是最终的地主。

2.4.2 出牌是否合法

用5位数存储牌型，包括(1位)牌型编号+(2位)关于张数的参数+(2位)牌的大小；00000表示跳过。

在mainWindow中存三个int成员 thisRoundCardA/B/C，游戏开始时均为0。

一方出牌，另外两方接到消息时，就修改其相应的thisRoundCard。

如果自己出牌时，发现另外两人的thisRoundCard均为0，那么自己拥有牌权。

牌型	编号	参数	大小
单张	1	无	0-12表示牌的大小
一对	2	无	同上
三带	3	牌的张数	同上，记录三条的大小
顺子	4	牌的张数	同上，记录最小的牌大小
连对	5	牌的张数	同上，记录最小的牌大小
四带二	6	牌的张数	同上，记录四条的大小
飞机	7	(1位)组数+(1位)0不带/1小翼/2大翼	同上，记录最小三条的大小
炸弹	8	无	同上
王炸	9	无	无

表格中的“无”用0占位。这样可以用5位数字表示出一切关于牌型的信息。

出牌时，先判断是否拥有牌权、特判炸弹和王炸的情况，然后用大小为13的sum数组储存每个大小的牌有几张。（这里大王和小王被存到一起了，但只需要后面判断单张比大小时特判一下就可以了）

如果有牌权：从9到1依次判断是否符合每一种牌型（首先看牌的总数是否符合），顺便记录上面写的关于牌型的5位信息，如果成功就返回。

如果无牌权，与前一组出的牌判断大小：还是从9到1判断，先判断是否符合牌型种类，再判断是否严格比上一家出的牌大。

