Poker

作者 王建楠

2019013296

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 Card Class Reference

The Card class 单张扑克牌 存储单张扑克牌的信息，以及对其信息的查询，和两张扑克牌的比较

```
#include <poker.h>
```

Public Member Functions

- Card ()=default
- Card (char, char, bool chosen=0)
- Card (const QString &)
- bool getChosen () const
- void reverse ()

  reverse 将 *chosen* 信息取反
- QString operator() () const

  operator () 扑克牌面信息
- void setChosen (bool)
- QString name () const

  name 查询该扑克对应的图片名称
- int isJoker () const
- bool operator< (const Card &A) const
- bool operator== (const Card &A) const
- bool operator!= (const Card &A) const
- Card operator+ (const int &k) const

Private Attributes

- char a [2]
- bool chosen

Friends

- QDebug operator<< (QDebug debug, const Card &A)

4.1.1 Detailed Description

The Card class 单张扑克牌 存储单张扑克牌的信息，以及对其信息的查询，和两张扑克牌的比较

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Card() [1/3]

```
Card::Card ( )  [default]
```

4.1.2.2 Card() [2/3]

```
Card::Card (
             char t1,
             char t2,
             bool chosen = 0 )
```

4.1.2.3 Card() [3/3]

```
Card::Card (
             const QString & s )
```

4.1.3 Member Function Documentation

4.1.3.1 getChosen()

```
bool Card::getChosen ( ) const
```

4.1.3.2 isJoker()

```
int Card::isJoker ( ) const
```

4.1.3.3 name()

```
QString Card::name ( ) const
```

name 查询该扑克对应的图片名称

4.1.3.4 operator"!=()

```
bool Card::operator!= (
             const Card & A ) const
```

4.1.3.5 operator()()

```
QString Card::operator() ( ) const
```

operator () 扑克牌面信息

4.1.3.6 operator+()

```
Card Card::operator+ (
             const int & k ) const
```

4.1.3.7 operator<()

```
bool Card::operator< (
             const Card & A ) const
```

4.1.3.8 operator==()

```
bool Card::operator== (
             const Card & A ) const
```

4.1.3.9 reverse()

```
void Card::reverse ( )
```

reverse 将 chosen 信息取反

4.1.3.10 setChosen()

```
void Card::setChosen (
             bool t )
```

4.1.4 Friends And Related Function Documentation

4.1.4.1 operator<<

```
QDebug operator<< (
             QDebug debug,
             const Card & A )  [friend]
```

4.1.5 Member Data Documentation

4.1.5.1 a

```
char Card::a[2]  [private]
```

4.1.5.2 chosen

```
bool Card::chosen  [private]
```

The documentation for this class was generated from the following files:

- /home/steaunk/THU/2020 Summer/Socket/poker/poker.h
- /home/steaunk/THU/2020 Summer/Socket/poker/poker.cpp

4.2 Game Class Reference

The Game class 斗地主游戏的游戏界面

`#include <game.h>`

Inheritance diagram for Game:

Collaboration diagram for Game:

Signals

- void sendMsg (QByteArray, int)
- void sendMsg (QByteArray)

Public Member Functions

- Game (int player, QMainWindow ∗, QWidget ∗parent=nullptr)
- ∼Game ()
- QByteArray typeMsg (int id)

  typeMsg 生成信息格式中的发信人和信息类型

Private Slots

- void dealMsg (QByteArray msg)

  dealMsg 处理信息 处理收到的信息，并按照信息类型执行相应的操作
- void on_callButton_clicked ()
- void on_uncallButton_clicked ()
- void on_playButton_clicked ()
- void on_passButton_clicked ()
- void on_restartButton_clicked ()
- void on_quitButton_clicked ()
- void paintEvent (QPaintEvent ∗)
- void mousePressEvent (QMouseEvent ∗)

Private Member Functions

- void init ()
- void startGame ()

  startGame 游戏准备完毕 向其他主机发送信息，表明该主机已准备完毕，当全部完毕时，由 *0* 号主机发牌及随机游戏顺序
- void beginGame ()

  beginGame 游戏开始
- void bidLandlord ()

  bidLandlord 开始叫地主
- void playCards ()

  playCards 出牌
- void newRound ()

  newRound 新一轮出牌开始
- void setUser (int)
- void endGame (int)

  endGame 游戏结束
- void newRoundClear ()

  newRoundClear 新一轮出牌时清除状态栏
- void Win ()

  Win 取得胜利 提示 *WIN* 以及播放胜利音乐
- void Lose ()

  Lose 失败 提示 *LOSE* 以及播放失败音乐

Private Attributes

- int player
- int cnt
- int landlord
- int player_p
- bool showLandlord
- bool isYourTurn = 0
- QMainWindow ∗ last
- Poker mycards
- Poker landlordcards
- Poker lastcards
- int cntCard [2]
- bool pass [3]
- QMediaPlaylist ∗ playlist
- QMediaPlayer ∗ mediaplayer
- PokerTool ∗ tool
- Ui::Game ∗ ui

4.2.1 Detailed Description

The Game class 斗地主游戏的游戏界面

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Game()

```
Game::Game (
              int player,
              QMainWindow ∗ last,
              QWidget ∗ parent = nullptr )  [explicit]
```

4.2.2.2 ∼Game()

```
Game::∼Game ( )
```

4.2.3 Member Function Documentation

4.2.3.1 beginGame()

```
void Game::beginGame ( )  [private]
```

beginGame 游戏开始

4.2.3.2 bidLandlord()

```
void Game::bidLandlord ( )  [private]
```

bidLandlord 开始叫地主

4.2.3.3 dealMsg

```
void Game::dealMsg (
             QByteArray msg )  [private], [slot]
```

dealMsg 处理信息 处理收到的信息，并按照信息类型执行相应的操作

4.2.3.4 endGame()

```
void Game::endGame (
             int pl )  [private]
```

endGame 游戏结束

4.2.3.5 init()

```
void Game::init ( )  [private]
```

4.2.3.6 Lose()

```
void Game::Lose ( )  [private]
```

Lose 失败 提示 LOSE 以及播放失败音乐

4.2.3.7 mousePressEvent

```
void Game::mousePressEvent (
            QMouseEvent * ev )  [private], [slot]
```

4.2.3.8 newRound()

```
void Game::newRound ( )  [private]
```

newRound 新一轮出牌开始

4.2.3.9 newRoundClear()

```
void Game::newRoundClear ( )  [private]
```

newRoundClear 新一轮出牌时清除状态栏

4.2.3.10 on_callButton_clicked

```
void Game::on_callButton_clicked ( )  [private], [slot]
```

4.2.3.11 on_passButton_clicked

```
void Game::on_passButton_clicked ( )  [private], [slot]
```

4.2.3.12 on_playButton_clicked

```
void Game::on_playButton_clicked ( )  [private], [slot]
```

4.2.3.13 on_quitButton_clicked

```
void Game::on_quitButton_clicked ( )  [private], [slot]
```

4.2.3.14 on_restartButton_clicked

```
void Game::on_restartButton_clicked ( )  [private], [slot]
```

4.2.3.15 on_uncallButton_clicked

```
void Game::on_uncallButton_clicked ( )  [private], [slot]
```

4.2.3.16 paintEvent

```
void Game::paintEvent (
             QPaintEvent *  )  [private], [slot]
```

4.2.3.17 playCards()

```
void Game::playCards ( )  [private]
```

playCards 出牌

4.2.3.18 sendMsg [1/2]

```
void Game::sendMsg (
             QByteArray ,
             int  )  [signal]
```

4.2.3.19 sendMsg [2/2]

```
void Game::sendMsg (
             QByteArray  )  [signal]
```

4.2.3.20 setUser()

```
void Game::setUser (
             int id )  [private]
```

4.2.3.21 startGame()

```
void Game::startGame ( )  [private]
```

startGame 游戏准备完毕 向其他主机发送信息，表明该主机已准备完毕，当全部完毕时，由 0 号主机发牌及随机游戏顺序

4.2.3.22 typeMsg()

```
QByteArray Game::typeMsg (
             int id )
```

typeMsg 生成信息格式中的发信人和信息类型

Parameters

id	信息类型

4.2.3.23 Win()

```
void Game::Win ( )  [private]
```

Win 取得胜利 提示 WIN 以及播放胜利音乐

4.2.4 Member Data Documentation

4.2.4.1 cnt

```
int Game::cnt  [private]
```

4.2.4.2 cntCard

```
int Game::cntCard[2]  [private]
```

4.2.4.3 isYourTurn

```
bool Game::isYourTurn = 0  [private]
```

4.2.4.4 landlord

```
int Game::landlord  [private]
```

4.2.4.5 landlordcards

```
Poker Game::landlordcards  [private]
```

4.2.4.6 last

```
QMainWindow* Game::last  [private]
```

4.2.4.7 lastcards

```
Poker Game::lastcards  [private]
```

4.2.4.8 mediaplayer

```
QMediaPlayer* Game::mediaplayer  [private]
```

4.2.4.9 mycards

```
Poker Game::mycards  [private]
```

4.2.4.10 pass

```
bool Game::pass[3]  [private]
```

4.2.4.11 player

```
int Game::player [private]
```

4.2.4.12 player_p

```
int Game::player_p [private]
```

4.2.4.13 playlist

```
QMediaPlaylist* Game::playlist [private]
```

4.2.4.14 showLandlord

```
bool Game::showLandlord [private]
```

4.2.4.15 tool

```
PokerTool* Game::tool [private]
```

4.2.4.16 ui

```
Ui::Game* Game::ui [private]
```

The documentation for this class was generated from the following files:

- /home/steaunk/THU/2020 Summer/Socket/poker/game.h
- /home/steaunk/THU/2020 Summer/Socket/poker/game.cpp

4.3 MainWindow Class Reference

The MainWindow class 斗地主游戏的准备界面

```
#include <mainwindow.h>
```

Inheritance diagram for MainWindow:

Collaboration diagram for MainWindow:

Signals

- void startConnection (int)

Public Member Functions

- MainWindow (int argc, char ∗argv[ ], QWidget ∗parent=nullptr)
- ∼MainWindow ()

Public Attributes

- int player

Private Slots

- void on_connectButton_clicked ()
- void ready ()

Private Attributes

- Ui::MainWindow ∗ ui
- QButtonGroup ∗ btnGroup
- Server ∗ playerConnect
- Game ∗ game = nullptr

4.3.1 Detailed Description

The MainWindow class 斗地主游戏的准备界面

4.3.2 Constructor & Destructor Documentation

4.3.2.1 MainWindow()

```
MainWindow::MainWindow (
              int argc,
              char ∗ argv[ ],
              QWidget ∗ parent = nullptr )
```

4.3.2.2 ∼MainWindow()

```
MainWindow::∼MainWindow ( )
```

4.3.3 Member Function Documentation

4.3.3.1 on_connectButton_clicked

```
void MainWindow::on_connectButton_clicked ( )  [private], [slot]
```

4.3.3.2 ready

```
void MainWindow::ready ( )  [private], [slot]
```

4.3.3.3 startConnection

```
void MainWindow::startConnection (
             int   )  [signal]
```

4.3.4 Member Data Documentation

4.3.4.1 btnGroup

```
QButtonGroup* MainWindow::btnGroup  [private]
```

4.3.4.2 game

```
Game* MainWindow::game = nullptr  [private]
```

4.3.4.3 player

```
int MainWindow::player
```

4.3.4.4 playerConnect

```
Server* MainWindow::playerConnect  [private]
```

4.3.4.5 ui

```
Ui::MainWindow* MainWindow::ui  [private]
```

The documentation for this class was generated from the following files:

- /home/steaunk/THU/2020 Summer/Socket/poker/mainwindow.h
- /home/steaunk/THU/2020 Summer/Socket/poker/mainwindow.cpp

4.4 MyTcp Class Reference

The MyTcp class 组合了服务端和客户端

```
#include <server.h>
```

Inheritance diagram for MyTcp:

Collaboration diagram for MyTcp:

Public Slots

- void acceptConnection ()
- void recvMsg ()

Public Member Functions

- MyTcp (Server ∗parent=nullptr)
- void startClient (QString ip, int port, QString playername)

  startClient 开启客户端
- void startServer (int port)

  startServer 开启服务端
- void write (QByteArray)

  write 发送信息

Private Attributes

- Server ∗ parent
- QTcpServer ∗ l
- QTcpSocket ∗ s

4.4.1 Detailed Description

The MyTcp class 组合了服务端和客户端

4.4.2 Constructor & Destructor Documentation

4.4.2.1 MyTcp()

```
MyTcp::MyTcp (
             Server * parent = nullptr )
```

4.4.3 Member Function Documentation

4.4.3.1 acceptConnection

```
void MyTcp::acceptConnection ( )  [slot]
```

4.4.3.2 recvMsg

```
void MyTcp::recvMsg ( )  [slot]
```

4.4.3.3 startClient()

```
void MyTcp::startClient (
             QString ip,
             int port,
             QString playername )
```

startClient 开启客户端

Parameters

ip	服务器ip地址
port	服务器网络端口
playername	服务器对象名称

4.4.3.4 startServer()

```
void MyTcp::startServer (
             int port )
```

startServer 开启服务端

Parameters

port	服务器网络端口

4.4.3.5 write()

```
void MyTcp::write (
             QByteArray msg )
```

write 发送信息

4.4.4 Member Data Documentation

4.4.4.1 l

```
QTcpServer* MyTcp::l  [private]
```

4.4.4.2 parent

```
Server* MyTcp::parent  [private]
```

4.4.4.3 s

```
QTcpSocket* MyTcp::s  [private]
```

The documentation for this class was generated from the following files:

- /home/steaunk/THU/2020 Summer/Socket/poker/server.h
- /home/steaunk/THU/2020 Summer/Socket/poker/server.cpp

4.5 Poker Class Reference

The Poker class 一组扑克 按照大小顺序存储扑克牌，可进行添加、移除、选择、输出信息

```
#include <poker.h>
```

Public Member Functions

- Poker ()=default
- Poker (std::vector< Card > cards)
- void addCard (const Card &A)
- void addCards (const std::vector< Card > &cards)
- int cntCards () const

  cntCards 统计牌数
- void removeCards (const std::vector< Card > &cards)
- void setAllChosen (bool d=false)
- void revCard (int id)
- bool getChosen (int id) const
- Poker chosenCards ()

  chosenCards 取出所有被选中的牌
- std::vector< Card > Cards () const
- void clear ()
- QByteArray write ()

  write 查询该牌组所有扑克的信息
- bool operator< (const Poker &) const
- Card operator[ ] (const int &id) const

Private Attributes

- std::vector< Card > cards

Friends

- QDebug operator<< (QDebug debug, const Poker &A)

4.5.1 Detailed Description

The Poker class 一组扑克 按照大小顺序存储扑克牌，可进行添加、移除、选择、输出信息

4.5.2 Constructor & Destructor Documentation

4.5.2.1 Poker() [1/2]

```
Poker::Poker ( )  [default]
```

4.5.2.2 Poker() [2/2]

```
Poker::Poker (
             std::vector< Card > cards )
```

4.5.3 Member Function Documentation

4.5.3.1 addCard()

```
void Poker::addCard (
             const Card & A )
```

4.5.3.2 addCards()

```
void Poker::addCards (
             const std::vector< Card > & cards )
```

4.5.3.3 Cards()

```
std::vector< Card > Poker::Cards ( ) const
```

4.5.3.4 chosenCards()

```
Poker Poker::chosenCards ( )
```

chosenCards 取出所有被选中的牌

4.5.3.5 clear()

```
void Poker::clear ( )
```

4.5.3.6 cntCards()

```
int Poker::cntCards ( ) const
```

cntCards 统计牌数

4.5.3.7 getChosen()

```
bool Poker::getChosen (
            int id ) const
```

4.5.3.8 operator<()

```
bool Poker::operator< (
            const Poker & A ) const
```

4.5.3.9 operator[]()

```
Card Poker::operator[ ] (
            const int & id ) const
```

4.5.3.10 removeCards()

```
void Poker::removeCards (
            const std::vector< Card > & cards )
```

4.5.3.11 revCard()

```
void Poker::revCard (
            int id )
```

4.5.3.12 setAllChosen()

```
void Poker::setAllChosen (
            bool d = false )
```

4.5.3.13 write()

```
QByteArray Poker::write ( )
```

write 查询该牌组所有扑克的信息

Returns

4.5.4 Friends And Related Function Documentation

4.5.4.1 operator<<

```
QDebug operator<< (
             QDebug debug,
             const Poker & A )  [friend]
```

4.5.5 Member Data Documentation

4.5.5.1 cards

```
std::vector<Card> Poker::cards  [private]
```

The documentation for this class was generated from the following files:

- /home/steaunk/THU/2020 Summer/Socket/poker/poker.h
- /home/steaunk/THU/2020 Summer/Socket/poker/poker.cpp

4.6 PokerTool Class Reference

The PokerTool class 扑克游戏工具箱 集合了一些进行扑克游戏需要的工具，如：发牌、判断是否可以出牌等

```
#include <poker.h>
```

Collaboration diagram for PokerTool:

Public Member Functions

- PokerTool ()
- void deal ()

  deal 发牌 按照斗地主规则发牌，随机选出 *17*、*17*、*17*、*3* 共四份牌，存储在 *poker[]* 中
- Poker read (const QByteArray &)

  read 读取卡牌 根据获得的字符串，识别读取到的扑克
- Poker getPoker (int)

  getPoker 发牌后，获取相应的牌叠
- QString check (const Poker &) const

  check 判断给定牌是否合法 按照斗地主规则判断。返回 *"F"* 则为不合法；反之，共返回长度为四的字符串，前两位为牌型，后两位为大小
- bool canPlay (const Poker &A, const Poker &B) const

  canPlay 判断给定牌是否可打出 按照斗地主规则判断。当上家打出一定牌 *B* 时，判断能否打出 *A*
- std::vector< Poker > sortPoker (const Poker &) const

Private Member Functions

- QString single (const Poker &) const
- QString rocket (const Poker &) const
- QString pair (const Poker &) const
- QString bomb (const Poker &) const
- QString triplet (const Poker &) const
- QString tripletWithSingle (const Poker &) const
- QString tripletWithPair (const Poker &) const
- QString sequence (const Poker &) const
- QString sequenceOfPairs (const Poker &) const
- QString sequenceOfTriplets (const Poker &) const
- QString sequenceOfTripletsWithSingle (const Poker &) const
- QString sequenceOfTripletsWithPair (const Poker &) const
- QString quadplexSet (const Poker &) const

Private Attributes

- std::vector< Card > allcards

  一副扑克牌所有牌的集合
- Poker pokers [4]
- std::vector< QString(PokerTool::∗)(const Poker &) const > funcSet

4.6.1 Detailed Description

The PokerTool class 扑克游戏工具箱 集合了一些进行扑克游戏需要的工具，如：发牌、判断是否可以出牌等

4.6.2 Constructor & Destructor Documentation

4.6.2.1 PokerTool()

```
PokerTool::PokerTool ( )
```

4.6.3 Member Function Documentation

4.6.3.1 bomb()

```
QString PokerTool::bomb (
             const Poker & t ) const  [private]
```

4.6.3.2 canPlay()

```
bool PokerTool::canPlay (
             const Poker & A,
             const Poker & B ) const
```

canPlay 判断给定牌是否可打出 按照斗地主规则判断。当上家打出一定牌 B 时，判断能否打出 A

Parameters

A	此时己方打出的牌
B	上家打出的牌，若为 "" 则无上家

4.6.3.3 check()

```
QString PokerTool::check (
             const Poker & t ) const
```

check 判断给定牌是否合法 按照斗地主规则判断。返回 "F" 则为不合法；反之，共返回长度为四的字符串，前两位为牌型，后两位为大小

Returns

- "F" Fail
- "Ro" Rocket
- "Si" Single Card
- "Pa" Pair
- "Bo" Bomb
- "Tr" Triplet

- "TS" Triplet with Single Card
- "TP" Triplet with Pair
- "Se" Sequence of Single Cards
- "SP" Sequence of Pairs
- "ST" Sequence of Triplets
- "S1" Sequence of Triplets with Single Card
- "S2" Sequence of Triplets with Pairs
- "QS"/"QP" Quadplex Set

4.6.3.4 deal()

```
void PokerTool::deal ( )
```

deal 发牌 按照斗地主规则发牌，随机选出 17、17、17、3 共四份牌，存储在 poker[] 中

4.6.3.5 getPoker()

```
Poker PokerTool::getPoker (
             int id )
```

getPoker 发牌后，获取相应的牌叠

4.6.3.6 pair()

```
QString PokerTool::pair (
             const Poker & t ) const  [private]
```

4.6.3.7 quadplexSet()

```
QString PokerTool::quadplexSet (
             const Poker & t ) const  [private]
```

4.6.3.8 read()

```
Poker PokerTool::read (
             const QByteArray & A )
```

read 读取卡牌 根据获得的字符串，识别读取到的扑克

4.6.3.9 rocket()

```
QString PokerTool::rocket (
             const Poker & t ) const  [private]
```

4.6.3.10 sequence()

```
QString PokerTool::sequence (
             const Poker & t ) const  [private]
```

4.6.3.11 sequenceOfPairs()

```
QString PokerTool::sequenceOfPairs (
             const Poker & t ) const  [private]
```

4.6.3.12 sequenceOfTriplets()

```
QString PokerTool::sequenceOfTriplets (
             const Poker & t ) const  [private]
```

4.6.3.13 sequenceOfTripletsWithPair()

```
QString PokerTool::sequenceOfTripletsWithPair (
             const Poker & t ) const  [private]
```

4.6.3.14 sequenceOfTripletsWithSingle()

```
QString PokerTool::sequenceOfTripletsWithSingle (
             const Poker & t ) const  [private]
```

4.6.3.15 single()

```
QString PokerTool::single (
             const Poker & t ) const  [private]
```

4.6.3.16 sortPoker()

```
std::vector< Poker > PokerTool::sortPoker (
             const Poker & t ) const
```

4.6.3.17 triplet()

```
QString PokerTool::triplet (
             const Poker & t ) const  [private]
```

4.6.3.18 tripletWithPair()

```
QString PokerTool::tripletWithPair (
             const Poker & t ) const  [private]
```

4.6.3.19 tripletWithSingle()

```
QString PokerTool::tripletWithSingle (
             const Poker & t ) const  [private]
```

4.6.4 Member Data Documentation

4.6.4.1 allcards

```
std::vector<Card> PokerTool::allcards  [private]
```

一副扑克牌所有牌的集合

4.6.4.2 funcSet

```
std::vector<QString(PokerTool::*)(const Poker &) const> PokerTool::funcSet  [private]
```

4.6.4.3 pokers

```
Poker PokerTool::pokers[4]  [private]
```

The documentation for this class was generated from the following files:

- /home/steaunk/THU/2020 Summer/Socket/poker/poker.h
- /home/steaunk/THU/2020 Summer/Socket/poker/poker.cpp

4.7 Server Class Reference

The Server class 斗地主专用通信工具

```
#include <server.h>
```

Inheritance diagram for Server:

Collaboration diagram for Server:

Public Slots

- void sendMsg (const QByteArray &, int id=-1)

  sendMsg 发送信息

Signals

- void ready ()
- void dealMsg (QByteArray)

Public Member Functions

- Server (bool testmode=0)
- ∼Server ()
- void inputIP (QString &, int &, QString)

  inputIP 输入*IP*地址
- int inputPort (int def, QString msg)
- void countDown ()

  countDown 记录其他玩家数 达到 *2* 名玩家时，游戏开始

Private Slots

- void startConnection (int)

Private Attributes

- int cnt = 0
- bool testMode

  值为 *1* 时，不用输入 *ip* 参数；值为 *0* 时需输入 *ip* 参数
- MyTcp ∗ s1

  s1 下家；*s2* 上家
- MyTcp ∗ s2

4.7.1 Detailed Description

The Server class 斗地主专用通信工具

4.7.2 Constructor & Destructor Documentation

4.7.2.1 Server()

```
Server::Server (
            bool testmode = 0 )
```

4.7.2.2 ∼Server()

```
Server::∼Server ( )
```

4.7.3 Member Function Documentation

4.7.3.1 countDown()

```
void Server::countDown ( )
```

countDown 记录其他玩家数 达到 2 名玩家时，游戏开始

4.7.3.2 dealMsg

```
void Server::dealMsg (
            QByteArray  )  [signal]
```

4.7.3.3 inputIP()

```
void Server::inputIP (
             QString & ip,
             int & port,
             QString msg = "" )
```

inputIP 输入IP地址

4.7.3.4 inputPort()

```
int Server::inputPort (
             int def = 8888,
             QString msg = "" )
```

4.7.3.5 ready

```
void Server::ready ( )  [signal]
```

4.7.3.6 sendMsg

```
void Server::sendMsg (
             const QByteArray & msg,
             int id = -1 )  [slot]
```

sendMsg 发送信息

Parameters

id	0 发送给下家； 1 发送给上家； -1 发送给所有玩家

4.7.3.7 startConnection

```
void Server::startConnection (
             int player )  [private], [slot]
```

4.7.4 Member Data Documentation

4.7.4.1 cnt

```
int Server::cnt = 0  [private]
```

4.7.4.2 s1

```
MyTcp* Server::s1  [private]
```

s1 下家；s2 上家

4.7.4.3 s2

```
MyTcp * Server::s2  [private]
```

4.7.4.4 testMode

```
bool Server::testMode  [private]
```

值为 1 时，不用输入 ip 参数；值为 0 时需输入 ip 参数

The documentation for this class was generated from the following files:

- /home/steaunk/THU/2020 Summer/Socket/poker/server.h
- /home/steaunk/THU/2020 Summer/Socket/poker/server.cpp

Chapter 5

File Documentation

5.1 /home/steaunk/THU/2020 Summer/Socket/poker/game.cpp File Reference

```
#include "game.h"
```
Include dependency graph for game.cpp:

5.2 /home/steaunk/THU/2020 Summer/Socket/poker/game.h File Reference

```
#include "ui_game.h"
#include "poker.h"
#include <QMainWindow>
#include <QKeyEvent>
#include <QMediaPlaylist>
#include <QMediaPlayer>
#include <QWidget>
```
Include dependency graph for game.h: This graph shows which files directly or indirectly include this file:

Classes

- class Game

  The Game class 斗地主游戏的游戏界面

Namespaces

- Ui

5.3 /home/steaunk/THU/2020 Summer/Socket/poker/main.cpp File Reference

```
#include "mainwindow.h"
#include <QDebug>
#include <QApplication>
```
Include dependency graph for main.cpp:

Functions

- int main (int argc, char ∗argv[ ])

5.3.1 Function Documentation

5.3.1.1 main()

```
int main (
            int argc,
            char * argv[] )
```

5.4 /home/steaunk/THU/2020 Summer/Socket/poker/mainwindow.cpp File Reference

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
```

Include dependency graph for mainwindow.cpp:

5.5 /home/steaunk/THU/2020 Summer/Socket/poker/mainwindow.h File Reference

```
#include <QButtonGroup>
#include <QMainWindow>
#include <QDebug>
#include <QtNetwork>
#include "server.h"
#include "game.h"
```

Include dependency graph for mainwindow.h: This graph shows which files directly or indirectly include this file:

Classes

- class MainWindow

  The MainWindow class 斗地主游戏的准备界面

Namespaces

- Ui

5.6 /home/steaunk/THU/2020 Summer/Socket/poker/poker.cpp File Reference

```
#include "poker.h"
```

Include dependency graph for poker.cpp:

5.7 /home/steaunk/THU/2020 Summer/Socket/poker/poker.h File Reference

主要负责处理扑克牌、扑克牌叠和一些对扑克的操作

```
#include <QString>
#include <vector>
#include <algorithm>
#include <QTime>
#include <QDebug>
#include <functional>
#include <QPainter>
```

Include dependency graph for poker.h: This graph shows which files directly or indirectly include this file:

Classes

- class Card

  The Card class 单张扑克牌 存储单张扑克牌的信息，以及对其信息的查询，和两张扑克牌的比较
- class Poker

  The Poker class 一组扑克 按照大小顺序存储扑克牌，可进行添加、移除、选择、输出信息
- class PokerTool

  The PokerTool class 扑克游戏工具箱 集合了一些进行扑克游戏需要的工具，如：发牌、判断是否可以出牌等

5.7.1 Detailed Description

主要负责处理扑克牌、扑克牌叠和一些对扑克的操作

Author

Wang Jiannan

Date

2020-9-02

5.8 /home/steaunk/THU/2020 Summer/Socket/poker/server.cpp File Reference

```
#include "server.h"
```

Include dependency graph for server.cpp:

5.9 /home/steaunk/THU/2020 Summer/Socket/poker/server.h File Reference

主要负责服务器、客户端连接通信

```
#include <QtNetwork>
#include <QMessageBox>
#include <QObject>
#include <QDebug>
#include <QInputDialog>
#include <QRegExp>
#include <vector>
```

Include dependency graph for server.h: This graph shows which files directly or indirectly include this file:

Classes

- class MyTcp

  The MyTcp class 组合了服务端和客户端
- class Server

  The Server class 斗地主专用通信工具

5.9.1 Detailed Description

主要负责服务器、客户端连接通信

Author

Wang Jiannan

Date

2020-9-02

Index