《谁说谎》程序说明文档

软件1207 王凌翔 20124862

1. 游戏简介

该游戏是一个规则较为简单的博弈游戏。在一副牌[[1]](#footnote-1)的情况下，推荐玩家人数2到6人。是一个比较适合聚会时，亲友之间所一起玩的游戏。

1. 游戏规则
   1. 在游戏发牌时，应保证每个玩家的手牌（up）一致。如果出现牌数(cp)不能被当前未拿手牌人数(n)整除，则采用以下算法[[2]](#footnote-2)：

例如：玩家数量为5的情况下，按照该算法计算：

第一名玩家牌数 = trunc(52 ÷ 5) = 10

第二名玩家牌数 = trunc(42 ÷ 4) = 10

第三名玩家牌数 = trunc(32 ÷ 3) = 10

第四名玩家牌数 = trunc(22 ÷ 2) = 11

第五名玩家牌数 = trunc(11 ÷ 1) = 11

所以每个人应该拿的牌数是：10、10、10、11、11.

* 1. 游戏一个回合分为以下几个阶段：

1. 出牌：一个玩家出牌，牌数任意。并倒置在桌面上。
2. 声明：该玩家必须声明他所出的牌的数量和点数。数量必须与所出牌的数量一致。只能声明一种点数，但是点数可以与所倒置牌的点数不一致。例如玩家倒置了2张A。他可在声明中说明：“我出了2张3”。但是类似这样的声明是不允许的：“我出了1个A和1个2”。
3. 判定：在回合中，只要有玩家声明，其他玩家都可以对该玩家的声明进行判定。若有玩家认为出牌玩家的声明为假，则可以要求翻牌查看。若出牌玩家所出牌与声明一致，则要求判定的玩家须拿走该回合所有所出之牌，如果不一致，则出牌玩家拿走该回合所有所出之牌。
4. 加牌：若其他玩家认为出牌玩家为真，则可以进行加牌。加牌时仍遵循着“出牌——声明——判定”的顺序。但是声明时，只声明牌数。点数默认为该回合第一个玩家所声明点数。例如，在该回合中，第一位玩家声明其所处牌的点数为2且无人判定为假，之后的玩家若加了2张牌，则声明时认为加了2张2，然后进入判定。
5. 回合结束：若没有玩家进行判定或者没有玩家进行加牌时，该回合结束。由下一名玩家进行出牌。
6. 程序流程图：

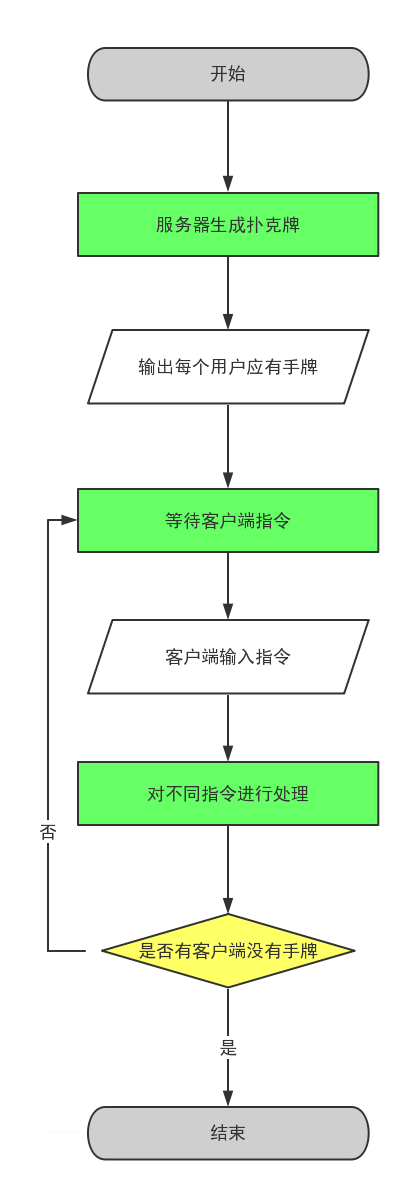
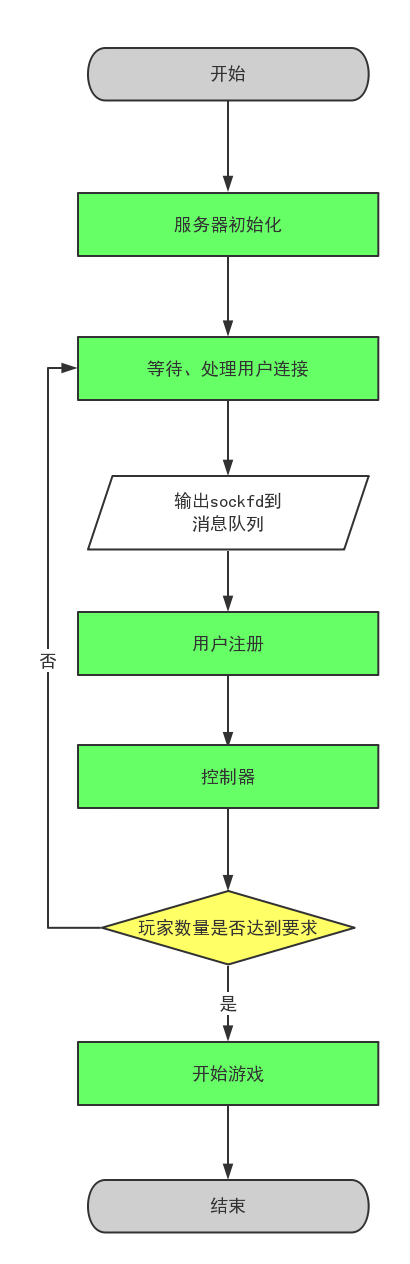


图1 程序初始化流程图

图2 游戏运行流程图

1. 客户端服务器通信协议：
   1. 客户端与服务器之间建立IPV4/TCP 连接。
   2. 扑克牌用一个字符代替，不考虑花色。数字2至9用所对应的字符代替。数字1用字符“A”代替，10用字符“+”代替，11用字符“J”代替，12用字符“Q”代替，13用字符“K”代替。
   3. 服务器采用以下语句与客户端进行通信：
      1. FORCE [pokers]

说明：服务器发牌给客户端（包括前期手牌和判定失败产生的手牌）

* + 1. TURN

说明：服务器要求客户端进行出牌操作

* + 1. JUDGE

说明：服务器要求客户端进行判定操作

* 1. 客户端采用以下语句与服务器进行通信：
     1. PUT [pokers]

说明：客户端进行出牌

* + 1. CLAIM [pokers]

说明：客户端进行声明

* + 1. SAY TRUE/FALSE

说明：客户端进行判定真/假

* + 1. OVER

说明：客户端声明已没有手牌（游戏开始之后）

1. 已编写服务器程序说明：
   * 1. 概述：采用两个线程进行处理。一个线程connecter负责维持与客户端之间的连接，并将客户端的指令存入消息队列[[3]](#footnote-3)msqid中；一个线程controler负责控制游戏流程。
     2. 仍未实现的功能模块：加牌。
     3. 程序所使用的一些重要数据结构说明：
        1. 消息队列con\_msqid负责传输客户端的sockfd
        2. 消息队列msqid负责传输客户端的指令
     4. 在通常情况下，msg.mtype代表sockfd，msg.mtext代表客户端所发送的指令。sockfd也默认作为userid.
     5. 服务器采用轮询的方法来让模拟每个玩家轮流出牌。
     6. 未来可以增加日志存储功能，现在暂时以printf输出。

1. 在本游戏中，一副牌的数量为52张，不包括大小Joker。欲增强趣味性，可引入大小Joker作为任意点数牌。 [↑](#footnote-ref-1)
2. 该计算方法保证任意两人之间牌数相差最小。但实际上，前期手牌数量的多少对游戏的公平不会产生很大影响。 [↑](#footnote-ref-2)
3. 这里所指的消息队列均为该消息队列的id [↑](#footnote-ref-3)