|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type（1字节） | len（2字节） | data（不定长） |
|  |  |  |  |

客户端向服务器发送的 登录报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
| 0x01 |  | data的长度 | 用户名 |

服务器向客户端发送的 确认登录成功的报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x51 | 0 |  |

服务器向客户端发送的 登录失败的报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x52 | 1 | 登录失败的原因 |

客户端向服务器发送 请求用户信息的报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
| 0x02 |  | 0 |  |

服务器向客户端发送 用户信息的报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x53 | data的长度 | 用户的信息 |

客户端发送请求 当前牌局的报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
| 0x03 |  | 0 |  |

服务器返回 当前牌局信息的报文（服务器开8个牌桌）

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x54 | data的长度 | 当前8个牌桌的信息 |

客户端发送 申请加入某个牌桌的报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
| 0x04 |  | data的长度 | 牌桌的编号 |

服务器发送 允许加入某个牌桌的报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x55 | data的长度 | 牌桌的编号 |

服务器发送 拒绝加入某个牌桌的报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x56 | data的长度 | 拒绝的原因 |

一个牌桌（共三个人），每加入一个人，服务器向牌桌每个发送当前牌桌的人员信息

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x57 | data的长度 | 当前游戏中牌桌的信息 |

#客户端收到后显示人员信息，并提示玩家选择明牌开始或开始

客户端发送 开始信息

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
| 0x05 |  | data的长度 | 玩家开始选项 |

#当服务器接收到三个玩家都准备好，服务器开始本局，并计算分的倍数

服务器向每个玩家发送随机的17张牌

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x58 | data的长度 | 17张牌 |

#如果有1个玩家选择了明牌开始，那么服务器首先询问这个玩家是否叫地主，如果有2，3个玩家选择明牌开始或者，服务器随机从一个明牌开始的玩家开始询问是否叫地主，如果没有一个玩家叫地主，就随机从一个玩家开始询问是否叫地主。

服务器发送 询问是否叫地主信息（服务器等待5秒，无反应默认不叫，并向下一个玩家发送，如果三个人都不叫，就重新向三个玩家发17张牌，并重新询问三个玩家是否叫地主）

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x59 | 0 |  |

#客户端接收到后依次显示17张排序后的牌

#客户端收到是否叫地主信息后，提示玩家选择是否叫地主，并向服务端发送是否叫地主

客户端发送 是否叫地主

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
| 0x06 |  | 1 | 是否叫地主 |

#服务器选择出地主后，将谁是地主和剩下的三张牌发给每位玩家

服务器发送 剩下三张牌

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x58 | data的长度 | 3张牌 |

服务器向三个玩家发送 谁是地主

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x5a | data的长度 | 玩家名字 |

#服务器从地主开始，依次询问每个玩家是否出牌，并给出上个玩家出的牌和每个玩家的牌数，等待30秒后默认不出

服务器发送 询问出牌报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x5b | 0 |  |

服务器向每个玩家发送 上个玩家出的牌

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x58 | data的长度(如果上个玩家不出，len=0) | n张牌 |

#如果玩家是掉线重连的，他会请求服务器发送每个玩家剩余的牌数

客户端发送 请求当前牌桌每个玩家剩余牌数

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
| 0x0a |  | 0 |  |

服务器向玩家发送 当前每个玩家剩余的牌

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x5e | data的长度 | 每个玩家剩余的牌数 |

#客户端在接收到询问出牌报文后，提示用户三个选项，出牌，不出，提示，提示选项只需根据玩家上次出牌信息在玩家有的牌中选择能压住的牌，如果上次出的牌仍是本次玩家出的，就可以随便出。玩家超过30秒不选择，按钮消失或者变灰。

玩家出的牌是否能压住上次玩家出的牌服务器和客户端都要验证。

玩家出的牌是否符合规则服务器和客户端都要验证。

如果玩家出的牌有炸或者飞机，客户端负责显示一个特效，服务端计算本局倍数

客户端发送 出牌报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
| 0x07 |  | data的长度 | 玩家出的牌 |

客户端发送 不出报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
| 0x08 |  | 0 |  |

#当有一个玩家的牌出完后，服务器判断是哪个玩家胜利如果地主先出完牌，地主胜。如果任意一个农民先出完牌，农民胜。

然后服务器向每个玩家发送胜负信息，并根据胜负和倍数计算每个玩家得到的分数（可以为负数）。然后加入他的分数。

服务器向每个玩家发送 胜负报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x5c | data的长度 | 胜负信息 |

#服务器在发送完胜负信息报文后，就将本桌信息初始化，但三个玩家还是在本桌，等待10秒后，若不选择，就将这个玩家踢出去。

服务器向某个玩家发送 踢人报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x5d | 0 |  |

#客户端在收到胜负报文后，向玩家显示得分或者扣分分数，并给玩家三个选择 明牌开始，开始，退出，等待10秒后若不选择就退出房间，并向服务端请求牌局报文（上面有）

客户端发送 退出报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
| 0x09 |  | 0 |  |

客户端发送明牌开始，开始，报文（上面有）

/\*掉线或逃跑重连\*/

#玩家掉线重连后

客户端发送 请求当前牌桌信息

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
| 0x0c |  | 0 |  |

服务器返回

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x57 | data的长度 | 当前游戏中牌桌的信息 |

#客户端没25秒向服务器发送一个心跳信号，证明自己还在线

客户端发送 心跳信号

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
| 0x0b |  | 0 |  |

#如果服务器超过25秒没有收到用户的心跳信号，就认为用户掉线，服务器向剩下的两个人或者1个人发送某人掉线

服务器发送 掉线报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x5f | data的长度 | 用户名 |

#如果有用户逃跑，服务器向剩下的两个人或者1个人发送某人逃跑

服务器发送 逃跑报文

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type | len | data |
|  | 0x60 | data的长度 | 用户名 |

玩家0出牌

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type（1字节） | len（2字节） | data（不定长） |
|  | 0x70 | data的长度 | 牌 |

玩1出牌

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type（1字节） | len（2字节） | data（不定长） |
|  | 0x71 | data的长度 | 牌 |

玩家2出牌

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type（1字节） | len（2字节） | data（不定长） |
|  | 0x72 | data的长度 | 牌 |

发地主的牌

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Type（1字节） | len（2字节） | data（不定长） |
|  | 0x80 | data的长度 | 牌 |