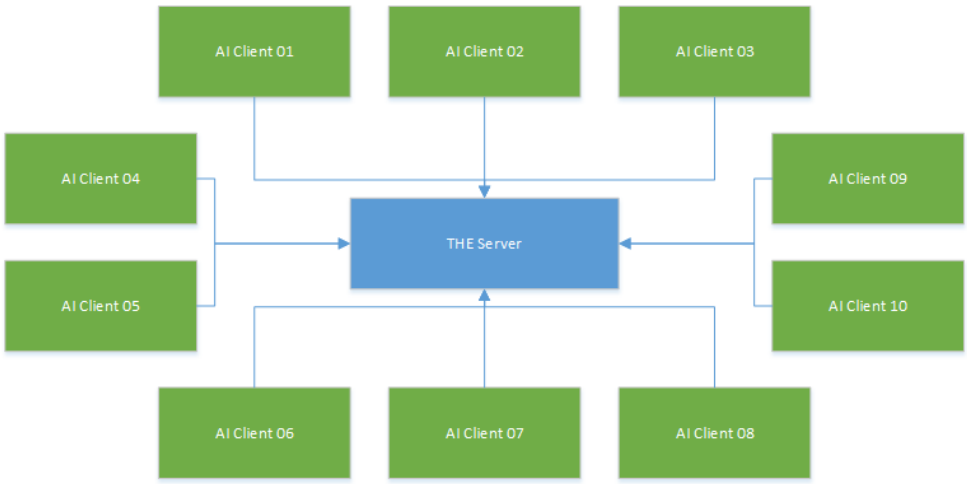
Texas Hold'em AI Game

文档 V1.0.7

2017-10-18

接入方式

本游戏引擎服务器端采用 Websocket 与 客户端相连接, 游戏为多人(上限为10人)对战模式, 至多允许10个客户端接入一个游戏桌 (game board)



Websocket 接入URI: "ws://serverIP:PORT"

测试接入(内外网都可访问) URI: "ws://pokerai.trendmicro.com.cn"

客户端在成功接入 Websocket 之后, 需要向服务器提交加入请求, 附带 playerName, playerName 在整个比赛中会唯一标识参赛玩家, 请妥善保存

客户端接入服务器引擎并开始游戏之后, 服务器将向客户端发送两种不同类型的消息

1. 通知客户端AI程序 "本次action应采取动作", 即服务器端告诉玩家 "本回合轮到你采取动作了", 此消息会在轮到某个客户端采取行动时发送, 为点对点消息, 此时客户端应通过 AI 决策程序计算出一种决策并发送给服务器

2. 通知客户端AI程序当前牌桌上的状态, 此消息在任意一名玩家有所行动或者牌局发生任何变化时发送, 为广播消息, 客户端需要以此更新本地保存的牌局状态, 以助于在轮到己方行动时采取最优决策

基本定义

行动说明和限制表 **(很重要, 很重要, 很重要)**

取值	定义
bet	下注, 以 amount 参数作为下注筹码金额 1. 如果 amount 小于当前游戏规定的最小押注额, 系统将自动转换为最小押注额 2. 如果玩家剩余筹码不够, 自动 allin 3. amount 字段如果不是数字, 那么系统自动进行弃牌操作 4. 金额必须是整数,如果不是整数,自动押注大盲注金额 5. 轮到强制 bet 的时候 (收到 __bet 指令) 也可以决策为 check/raise/allin 6. 如果超过Bet次数(每回合上限为 **4** 次), 系统将自动执行 call
call	跟注, 与前一位玩家押下相等的筹码 1. 如果玩家剩余的筹码不够, 系统将自动执行allin 2. 如果轮到玩家决策bet, 而发现玩家决策出call, 那么自动 bet 小盲注
raise	加注, 提高现有筹码为前一位玩家的2倍 1. 如果超过加注次数(每回合上限为 **4** 次), 系统将自动执行 call 2. 如果剩余筹码不够, 系统自动执行 allin 3. 如果应该决策出 bet, 而发现玩家决策出 raise, 那么系统自动 bet 大盲注
check	过牌, 在不需要跟注的情况下, 把押注机会传递给下一位玩家 1. 如果当前无法check的话 (牌桌上最大注额大于当前这名玩家的注额), 系统会自动执行call
fold	弃牌, 放弃本轮比赛的所有手牌
allin	全押, 将目前剩余的所有筹码(chips)作为注额

牌面值

1. 牌面统一由2个字符组成

2. 第1个字符为数字: 取值为: 2~9, T(10), J(J), Q(Q), K(K), A(Ace)

3. 第2个字符为花色: 分别为 H(红心), S(黑桃), C(草花), D(方块)

4. 例如: 3H -- 红心3; TS -- 黑桃10; QC -- 草花皇后; AD -- 方块Ace

加入游戏

加入游戏命令在游戏开始之前, 玩家进入游戏桌时由客户端向服务器发送, 命令格式 (JSON) 如下:

```
{
    "eventName" : "__join",
    "data" : {
        "playerName" : "0123456789ABCDEF" // 为了防止冒名作弊，这个名字是比赛系统预先分配给每个参赛玩家的私有名字，请玩家牢记并只在自己的程序中使用
                                          // 这个名字目前在测试平台上并不启用，在比赛前夕会通过私密的方式通告给每个参赛玩家
                                          // 每个玩家的私有名会对应一个展示名(由 Task Force事先录入)，比赛当日大屏幕上显示的为展示名以区分玩家
                                          // 以下文档中所有涉及到的 playerName 均为私有名，为了阅读方便，文档中暂时用类似 bobi，cicy，geoge 等拟名代替
                                          // 后台在收到玩家 __join 消息之后，将这个名字做 128bit 的 MD5 hash，
                                          // 生成一个 token，这个 token 将作为后续所有指令中的 playerName 出现，
                                          // 如果玩家希望在客户端代码中做相应的匹配，请自行在客户端中生成 128bit 的 MD5 hash，英文字母小写，也可以使用
                                          // 开发工具中的哈希工具来帮助生成并加入 hard code 到目标代码中
    }
}
```

指令说明:

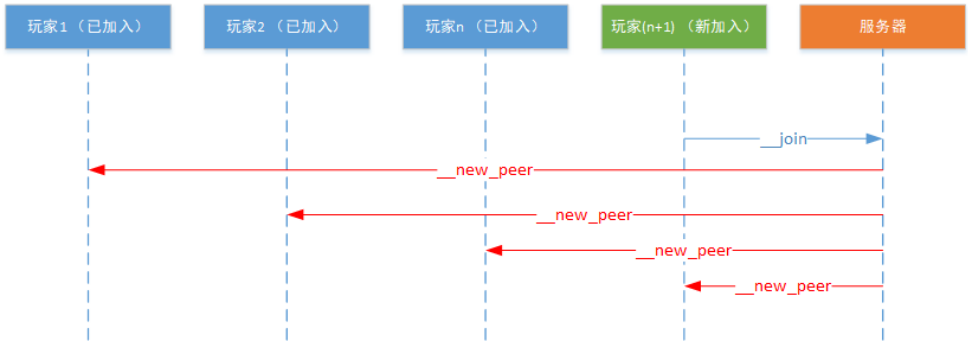
1. playerName 必须是 Task Force 赋予玩家的名字, 不可冒名顶替, 一旦被程序检查出冒名行为,将取消比赛资格

2. 如果加入成功, 那么服务器会将当前牌桌上所有玩家的列表向所有在线的客户端广播一次, 广播格式如下:

```
{
    "eventName" : "__new_peer",
    "data" : ["bobi", "cicy", "geoge", ... , "someone"],
}
```

3. __new_peer指令中的data包含在场所有玩家的名称列表

加入的流程如下:



新一轮开始

游戏开始时, 客户端都将收到服务器广播的 __new_round 消息

```
{
    "eventName" : "__new_round",
    "data" : {
        "table": {
            "tableNumber": "1",
            "roundName": "Deal",
            "board": [],
            "roundCount": 1,
            "raiseCount" : 0,
            "betCount" : 0,
            "smallBlind": {
                "playerName": "bobi(MD5 Hash)",
                "amount": 10
            },
            "bigBlind": {
                "playerName": "cicy(MD5 Hash)",
                "amount": 20
            }
        },
        "players" : [
            {
                "playerName" : "bobi(MD5 Hash)",
                "chips" : 20,
                "folded" : true,
                "allIn" : false,
                "isSurvive" : true,
                "reloadCount" : 2,
                "roundBet" : 40,
                "bet" : 40,
                "cards": [] // 只有自己的手牌才可见
            },
            {
                "playerName" : "cicy(MD5 Hash)",
                "chips" : 0,
                "folded" : true,
                "allIn" : true,
                "isSurvive" : false,
                "reloadCount" : 2,
                "roundBet" : 40,
                "bet" : 40,
                "cards": [] // 只有自己的手牌才可见
            },
            {
                "playerName" : "geoge(MD5 Hash)",
                "chips" : 500,
                "folded" : true,
                "allIn" : false,
                "isSurvive" : true,
                "reloadCount" : 2,
                "roundBet" : 40,
                "bet" : 40,
                "cards": [] // 只有自己的手牌才可见
            },
            ...
        ]
    }
}
```

指令说明:

1. data.table 字段表示当前牌桌的状态, data.table.tableNumber 表示当前的桌号, 桌号是游戏 Task Force 在比赛之前按照分组分桌分好的一个桌号, 在比赛开始之前由 Task Force人员将玩家录入系统, 理论上某个玩家 (playerName) 所属的桌号是固定且唯一的, 玩家客户端在发起 __join 指令表示加入时, 通过指定 playerName, 就能自动加入事先安排好的游戏桌

2. data.table.roundName 表示当前牌局的状态, 有如下取值: Deal(还未发公共牌), Flop(第1次发3张公共牌), Turn(发第4张公共牌), River(发第5张公共牌), 详情请参考文档末尾附表

3. data.table.board 表示当前桌面上的公开牌的点数, 牌面取值请参考文档末尾附表

4. data.players 字段表示当前牌桌上所有玩家的状态, data.players[i].chips 代表当前这个玩家剩余的筹码

5. data.players[i].folded 表示玩家是否已经 fold, data.players[i].allIn 表示玩家是否已经 all in, data.players[i].isSurvive 表示当前玩家是否仍在局 (false 表示输光出局)

索取筹码

在每一轮(除第1轮外)开始的时候, 服务器会发送询问玩家是否要索取筹码, 每局比赛中每次可以索取 1000 筹码, 最多索取 2 次

服务器发送的请求格式如下:

```
{
    "eventName" : "__start_reload",
    "data" : {
        "players": [
            {
                "playerName": "bobi(MD5 Hash)",
                "chips": 960,
                "folded": false,
                "": false,
                "isSurvive": true,
                "reloadCount": 0
            },
            {
                "playerName": "cicy(MD5 Hash)",
                "chips": 960,
                "folded": false,
                "allIn": false,
                "isSurvive": true,
                "reloadCount": 0
            },
            {
                "playerName": "geoge(MD5 Hash)",
                "chips": 960,
                "folded": false,
                "allIn": false,
                "isSurvive": true,
                "reloadCount": 0
            },
            ...
        ],
        "tableNumber": "1"
    }
}
```

指令说明:

1. 这条命令中包含当前所有玩家的剩余筹码以及已经索取筹码的次数

2. 如果你需要索取筹码, 这条命令需要在 **3** 秒内回复, 如果不回复, 那么服务器将自动忽略, 但是如果当前你已经没有筹码, 且索取次数在2次以内, 服务器将自动为你索取 1000 筹码

3. 回复指令如下:

```
{
    "eventName" : "__reload"
}
```

荷官发牌

在 Flop, Turn, River 发牌的时候服务器会广播这条命令, 格式如下:

```
{
    "eventName" : "__deal",
    "data" : {
        "players": [
            {
                "playerName": "bobi(MD5 Hash)",
                "chips": 980,
                "folded": false,
                "allIn": false,
                "isSurvive": true,
                "reloadCount": 0,
                "roundBet": 20,
                "bet": 0,
                "cards": []
            },
            {
                "playerName": "cicy(MD5 Hash)",
                "chips": 980,
                "folded": false,
                "allIn": false,
                "isSurvive": true,
                "reloadCount": 0,
                "roundBet": 20,
                "bet": 0,
                "cards": []
            },
            {
                "playerName": "geoge(MD5 Hash)",
                "chips": 980,
                "folded": false,
                "allIn": false,
                "isSurvive": true,
                "reloadCount": 0,
                "roundBet": 20,
                "bet": 0,
                "cards": []
            },
            ...
        ],
    "table": {
        "tableNumber": "1",
        "roundName": "Flop",
        "board": [
            "AC",
            "8S",
            "AH"
        ],
        "roundCount": 1,
        "raiseCount": 0,
        "betCount": 0,
        "smallBlind": {
            "playerName": "geoge(MD5 Hash)",
            "amount": 10
        },
        "bigBlind": {
            "playerName": "tryest(MD5 Hash)",
            "amount": 20
```

```
        }
    }
}
```

服务器请求玩家决策本次**action**应采取的动作(轮到某玩家行动)

比赛过程中, 服务器会在轮到需要某玩家客户端做出决策之际向该玩家客户端发送此消息(参考"通信机制.第1条"), 此消息又分为2类: 需要玩家行动和需要玩家押注

1. 需要玩家行动

```
{
    "eventName" : "__action",
    "data" : {
        "tableNumber" : 1,
        "self" : {
            "playerName" : "bobi(MD5 Hash)",
            "chips" : 1000,
            "folded" : false,
            "allIn" : false,
            "cards" : [XX, XX],
            "isSurvive" : true,
            "roundBet" : 0,
            "bet" : 10,
            "minBet" : 10,
            "cards" : ["JC", "KS"]
        },
        "game" : {
            "smallBlind": {
                "playerName" : "bobi(MD5 Hash)",
                "amount" : 10
            },
            "bigBlind": {
                "playerName" : "cicy(MD5 Hash)",
                "amount" : 20
            },
            "board" : [XX, XX, XX],
            "raiseCount" : 0,
            "betCount" : 1,
            "roundName" : "Deal", // 其它可能的值为 : "Flop", "Turn", "River"
            "players" : [
                {
                    "playerName" : "bobi(MD5 Hash)",
                    "chips" : 20,
                    "folded" : true,
                    "allIn" : false,
                    "isSurvive" : true,
                    "reloadCount" : 0,
                    "roundBet" : 0,
                    "bet" : 10,
                    "cards" : ["JC", "KS"] // 由于 self 也是 bobi, 所以在 player 列表里只
公开 bobi 的手牌
                },
                {
                    "playerName" : "cicy(MD5 Hash)",
                    "chips" : 0,
                    "folded" : true,
                    "allIn" : true,
                    "isSurvive" : false,
                    "reloadCount" : 0,
                    "roundBet" : 0,
                    "bet" : 10
                },
                {
                    "playerName" : "geoge(MD5 Hash)",
                    "chips" : 500,
                    "folded" : true,
                    "allIn" : false,
```

```
                    "isSurvive" : true,
                    "reloadCount" : 0,
                    "roundBet" : 0,
                    "bet" : 10
                },
                ...
            ]
        }
    }
}
```

指令说明:

1.0. "需要玩家行动" 是指的系统请求玩家决策一种行动, 可以是 "call", "check", "raise", "fold", "allin", "bet" 中的一种

1.1. data.self 字段表示当前己方(玩家自己)的状态, 除了 playerName, chips和几个状态之外, 通过这条信息玩家可以了解到自己的私有牌, 由于此消息是点对点发送, 因此玩家自己的私有牌不会泄露给其它玩家

1.2. data.game 表示当前牌局的状态, 其中 data.game.otherPlayers 字段中的牌桌上的所有玩家列表数据格式和内容含义均和 __new_round 指令中一致, 此处不再赘述

1.3. data.game.board 字段表示目前牌桌上的公共牌信息

1.4. data.self.minBet 字段表示当前可以押注的最小值

1.5. data.game.raiseCount 字段表示本回合中所有玩家总共加注 (raise) 的次数

1.6. data.game.betCount 字段表示本回合中所有玩家总共下注 (bet) 的次数

1.7. data.game.roundName 字段表示牌局的当前状态, 有如下取值: Deal(还未发公共牌), Flop(第1次发3张公共牌), Turn(发第4张公共牌), River(发第5张公共牌), 详情请参考文档末尾附表

1.8.data.game.smallBlind 和 data.game.bigBlind 分别展示了小盲注和大盲注玩家的位置和注额

玩家客户端收到此条指令之后, 需要向服务器提交 __action 决策, action 的类型可以是 "call", "check", "fold", "allin", "raise", "bet" 中的一种, 决策的数据格式如下

```
{
    "eventName" : "__action",
    "data" : {
        "action" : "call" // 还可以是 : "check", "fold", "allin", "raise", "bet" 中的其中一种,最新规则中允许是 "bet", bet 中的 amount 注额加上玩家当前已下注额之后不得小于当前最大注额
    }
}
```

2. 需要玩家下注

```
{
    "eventName" : "__bet",
    "data" : {
        "tableNumber" : 1,
        "self" : {
            "playerName" : "bobi(MD5 Hash)",
            "chips" : 1000,
            "folded" : false,
            "allIn" : false,
            "cards" : [XX, XX],
            "isSurvive" : true
            "roundBet" : 0,
            "bet" : 10,
            "minBet" : 10,
        },
        "game" : {
            "smallBlind": {
                "playerName": "bobi(MD5 Hash)",
                "amount": 10
            },
            "bigBlind": {
                "playerName": "cicy(MD5 Hash)",
                "amount": 20
            },
            "board" : [],
            "raiseCount" : 0,
            "betCount" : 1,
            "roundName" : "Deal", // 其它可能的值为 : "Flop", "Turn", "River"
            "players" : [
                {
                    "playerName" : "bobi(MD5 Hash)",
                    "chips" : 20,
                    "folded" : true,
                    "allIn" : false,
                    "isSurvive" : true,
                    "reloadCount" : 0,
                    "roundBet" : 0,
                    "bet" : 10,
                    "cards" : ["JC", "KS"] // 由于 self 也是 bobi, 所以在 player 列表里只
公开 bobi 的手牌
                },
                {
                    "playerName" : "cicy(MD5 Hash)",
                    "chips" : 0,
                    "folded" : true,
                    "allIn" : true,
                    "isSurvive" : false,
                    "reloadCount" : 0,
                    "roundBet" : 0,
                    "bet" : 10
                },
                {
                    "playerName" : "geoge(MD5 Hash)",
                    "chips" : 500,
                    "folded" : true,
                    "allIn" : false,
                    "isSurvive" : true,
```

```
                    "reloadCount" : 0,
                    "roundBet" : 0,
                    "bet" : 10
                },
                ...
            ]
        }
    }
}
```

指令说明:

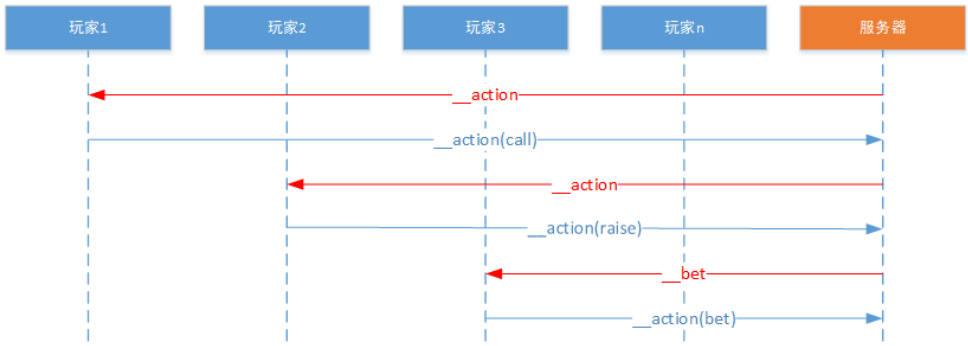
2.0. "需要玩家下注" 是指的系统请求玩家决策bet, 注意, 玩家客户端收到此消息之后仅可以决策bet, 但是可以选择 bet 的数值(amount)

2.1. 其它各字段含义和 __action 指令相同, 请参考 __action 指令说明

玩家客户端收到此条指令之后, 需要向服务器提交 __action 决策, action 的类型只能是 bet, 决策的数据格式如下, 其中包含本action下注的筹码

```
{
    "eventName" : "__action",
    "data" : {
        "action" : "bet", // 这里只能是"bet"
        "amount" : 100 // amount 在这里是一个强制字段
    }
}
```

服务器轮询以及玩家响应的流程如下:



服务器更新牌桌状态

比赛过程中, 服务器会根据牌桌上的任意状态变更(例如有玩家采取了某种行动) 广播牌桌状态给所有玩家客户端

```
{
    "eventName" : "__show_action",
    "data" : {
        "action" : {
            "action" : "call",
            "playerName" : "alex(MD5 Hash)",
            "amount" : 10,
            "chips" : 500
        },
        "table" : {
            "tableNumber" : 1,
            "roundName" : "Turn",
            "board" : [XX, XX, XX, XX],
            "roundCount": 1,
            "raiseCount" : 0,
            "betCount" : 1,
            "totalBet" : 2000,
            "smallBlind": {
                "playerName": "geoge(MD5 Hash)",
                "amount": 10
            },
            "bigBlind": {
                "playerName": "bobi(MD5 Hash)",
                "amount": 20
            }
        },
        "players" : [
            {
                "playerName" : "bobi(MD5 Hash)",
                "chips" : 20,
                "folded" : true,
                "allIn" : false,
                "isSurvive" : true,
                "cards" : [],
                "reloadCount" : 2,
                "roundBet" : 40,
                "bet" : 40
            },
            {
                "playerName" : "cicy(MD5 Hash)",
                "chips" : 0,
                "folded" : true,
                "allIn" : true,
                "isSurvive" : false,
                "cards" : [],
                "reloadCount" : 2,
                "roundBet" : 40,
                "bet" : 40
            },
            {
                "playerName" : "geoge(MD5 Hash)",
                "chips" : 500,
                "folded" : true,
                "allIn" : false,
                "isSurvive" : true,
                "cards" : [],
                "reloadCount" : 2,
```

```
                "roundBet" : 40,
                "bet" : 40
            },
            ...
        ]
    }
}
```

指令说明:

1. __show_action 是一条服务器的广播消息, 目的是通知所有玩家(不论是否出局), 当前牌桌的状态, 便于玩家AI程序根据当前状态做出决策

2. data.action 表示当前导致牌局发生变化的任意一名玩家的行动, data.action.amount 表示当前这名玩家所追加的赌注的数值(不论是call, raise还是bet)

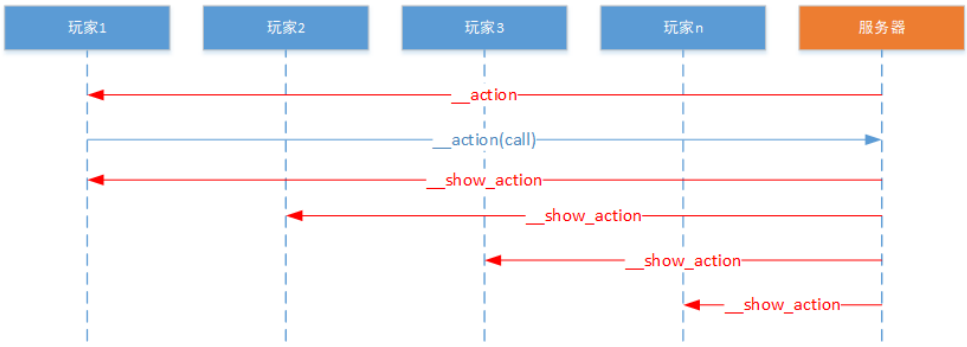
3. data.table 表示当前牌桌的状态, 包括公共牌的状态和已经发出的公共牌

4. data.players 表示当前所有玩家的状态, 请参考前序指令中的说明, 这里不再赘述

5. data.table.smallBlind 和 data.bigBlind 分别展示了小盲注和大盲注玩家的位置和注额

6. data.table.totalBet 表示当前牌桌底池的总筹码数额

以下是某次回合导致牌局发生变化之后服务器和客户端交互的消息示意图



每一轮结束

在每一轮结束的时候, 服务器会向所有玩家广播这一轮所有玩家的状态, 包括玩家手牌

```
{
    "eventName" : "__round_end",
    "data" : {
        "players": [
            {
                "playerName": "bobi(MD5 Hash)",
                "chips": 960,
                "folded": true,
                "allIn": false,
                "cards": [
                    "AD",
                    "3D"
                ],
                "isSurvive": true,
                "reloadCount": 0,
                "roundBet": 0,
                        "bet": 0,
                        "hand": {
                    "cards": ["AD",
                                    "3D",
                                    "9S",
                                    "9D",
                                    "2H",
                                    "AC",
                                    "9C"],
                                "rank": 219,
                                "message": "Full House"
                        },
                        "winMoney": 0,
                "isOnline": true
            },
            {
                "playerName": "cicy(MD5 Hash)",
                "chips": 1280,
                "folded": true,
                "allIn": false,
                "cards": [
                    "AD",
                    "TD"
                ],
                "isSurvive": true,
                "reloadCount": 0
            },
            ...
        ],
        "table": {
            "tableNumber": "1",
            "roundName": "Showdown",
            "board": [
                "AD",
                "4S",
                "2C",
                "9S",
                "5H"
            ],
            "roundCount": 1,
            "raiseCount" : 0,
```

```
            "betCount" : 1,
            "smallBlind": {
                "playerName": "bobi(MD5 Hash)",
                "amount": 10
            },
            "bigBlind": {
                "playerName": "cicy(MD5 Hash)",
                "amount": 20
            }
        }
    }
}
```

指令说明:

1. __round_end 是一条服务器的广播消息, 目的是让玩家知道本轮所有其他玩家的手牌以及公开牌, 为玩家准备下一轮策略做准备

2. data.player[i].cards 表示已经结束掉的这轮里的玩家的手牌

3. 这一条信息不需要玩家回复

游戏(一局)结束

游戏结束的时候, 服务器会向所有玩家广播以下消息

```
{
    "eventName" : "__game_over",
    "data" : {
        "players":
        [
            {
                "playerName": "bobi(MD5 Hash)",
                "chips": 0,
                "folded": false,
                "allIn": true,
                "cards": [
                    "3S",
                    "3C"
                ],
                "isSurvive": true,
                "reloadCount": 2,
                "reloadCount" : 2,
                "roundBet" : 40,
                "bet" : 40
            },
            {
                "playerName": "cicy(MD5 Hash)",
                "chips": 15900,
                "folded": true,
                "allIn": true,
                "cards": [
                    "AD",
                    "QH"
                ],
                "isSurvive": true,
                "reloadCount": 2,
                "reloadCount" : 2,
                "roundBet" : 40,
                "bet" : 40
            },
            {
                "playerName": "alex(MD5 Hash)",
                "chips": 0,
                "folded": true,
                "allIn": true,
                "cards": [
                    "JH",
                    "8D"
                ],
                "isSurvive": true,
                "reloadCount": 2,
                "reloadCount" : 2,
                "roundBet" : 40,
                "bet" : 40
            },
            ...
        ],
        "table": {
            "tableNumber": "1",
            "roundName": "Showdown",
            "board": [
                "AD",
```

```
                "4S",
                "2C",
                "9S",
                "5H"
            ],
            "roundCount": 16,
            "raiseCount" : 0,
            "betCount" : 1,
            "smallBlind": {
                "playerName": "bobi(MD5 Hash)",
                "amount": 10
            },
            "bigBlind": {
                "playerName": "cicy(MD5 Hash)",
                "amount": 20
            }
        },
        "winners": [
            {
                "playerName": "andy(MD5 Hash)",
                "hand": {
                    "cards": [
                        "AD",
                        "QH",
                        "8S",
                        "QC",
                        "QS",
                        "QD",
                        "2S"
                    ],
                    "rank": 290.2048,
                    "message": "Four of a kind"
                },
                "chips": 15900
            },
            {
                "playerName": "lance1(MD5 Hash)",
                "hand": {
                    "cards": [
                        "8H",
                        "KS",
                        "8S",
                        "QC",
                        "QS",
                        "QD",
                        "2S"
                    ],
                    "rank": 250,
                    "message": "Full House"
                },
                "chips": 1360
            },
            {
                "playerName": "tryest(MD5 Hash)",
                "hand": {
                    "cards": [
                        "9D",
                        "JD",
```

```
                        "8S",
                        "QC",
                        "QS",
                        "QD",
                        "2S"
                    ],
                    "rank": 110.032,
                    "message": "Three of a Kind"
                },
                "chips": 740
            }
        ]
    }
}
```

指令说明

1. __game_over 是一条服务器的广播消息, 这里表示比赛(预赛或者复赛或者决赛)结束, 玩家可以退出比赛

2. winners中的信息表示每个玩家最后一轮的手牌以及最终赢得的筹码数

3. 这一条信息不需要玩家回复

其它注意事项

1. 服务器在向本次action需要行动的玩家发送 __action 请求决策之后, 会等待 **5秒**, 决策超时时间为 **5秒**, 如果客户端出现异常, 掉线或者在超时时间之内无法做出决策, **服务器会对该玩家采取弃牌操作**

2. 如果玩家在决策时发送了错误或者规定以外的指令, **服务器会对该玩家采取弃牌操作**, 请谨慎编写决策程序

3. 没有轮到某玩家的action, 此玩家却采取了行动, 此行动将自动被忽略

4. 建议玩家客户端监听掉线事件,以便在掉线之后可以随时通过 Websocket 重新加入并继续游戏, 建议在掉线重连之后立即发送 __action, call

5. 每开始新的一圈时, 大小盲注会较之上一圈翻倍

6. 如果玩家在服务端发轮询动作之前离线, 系统会等待 **10秒**, **10秒** **后仍然没有等到再次接入, 将视为自动弃牌**