

森林舞会

- 协议文件名
- 游戏通用结构
  - 房间信息
  - 玩家信息
  - 玩家下注信息
- 协议列表
  - 检测游戏状态 //重连之后判断是否在桌子中
  - 请求桌子内场景信息
  - 离开桌子
  - 下注
  - 续压
  - 通知开始押注
  - 通知开奖
  - 所有玩家下注信息
  - 游戏控制
  - debug信息
  - 通知所有玩家下注信息
  - 提醒离开
- 协议ID枚举
- 游戏状态

# 森林舞会

## 协议文件名

game\_forestball\_protocol.proto

game\_forestball\_def.proto

## 游戏通用结构

### 房间信息

msg\_room\_info

参数名	类型	描述	参数ID	默认值
roomid	int32	房间Id	1	

### 玩家信息

PlayerInfo

参数名	类型	描述	参数ID	默认值
player_id	int32	玩家ID	1	
player_nickname	string	昵称	2	
player_head_frame	int32	头像框	3	
player_head_custom	string	玩家头像	4	
player_gold	int64	玩家金币	5	
player_sex	int32	玩家性别	6	
player_vip_lv	int32	Vip等级	7	

## 玩家下注信息

### msg\_bet\_info

参数名	类型	描述	参数ID	默认值
bet_index	int32	下注索引	1	
bet_count	int64	下注金额	2	

### gm\_msg\_bet\_info

参数名	类型	描述	参数ID	默认值
player_info	player_info	玩家信息	1	
bet_info	msg_bet_info[]	下注信息	2	

## 协议列表

### 检测游戏状态 //重连之后判断是否在桌子中

#### 前端请求协议:packetc2l\_check\_state

说明：登录游戏后调用，用来判断是否在游戏房间内，如果在，走断线重连逻辑直接进房间

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_c2l_check_state 10004

#### 后端返回协议:packetl2c\_check\_state\_result

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_l2c_check_state_result 15004
is_intable	bool	是否在房间中	2	

## 请求桌子内场景信息

说明：进入游戏后调用，获取游戏内的场景信息，用于客户端初始化游戏

前端请求协议:packetc2l\_get\_room\_scene\_info

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_c2l_get_room_scene_info 10003

后端返回协议:packetl2c\_get\_room\_scene\_info\_result

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_l2c_get_room_scene_info 15003
room_id	int32	房间id	2	
cd_time	int32	当前剩余时间	3	
room_state	e_game_state	游戏状态	4	
self_bet_list	int32[]	自己下注信息,断线重连用	5	
bet_info_list	PokerInfo[]	下注信息	6	
history_list	int32[]	历史牌路信息	7	
rate_index	int32	赔率ID	8	
colors	int32[]	24个数值 0:红 1:绿 2:黄	9	

## 离开桌子

说明：目前客户端离开房间调用统一的回到大厅方法，此此方法暂时不用

前端请求协议:packetc2l\_leave\_room

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_c2l_leave_room 10002

后端返回协议:packetl2c\_leave\_room\_result

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_l2c_leave_room_result 15002
result	msg_type_def.e_msg_result_def	结果	2	1.成功 非1.失败
player_gold	int64	玩家金币	3	

## 下注

说明：在游戏状态进入到下注阶段时可以调用

前端请求协议:packetc2l\_add\_bet

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_c2l_add_bet 10005
bet_index	int32	下注索引	2	
bet_count	int32	下注金额	3	

后端返回协议:packetl2c\_add\_bet\_result

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_l2c_add_bet 15005
result	msg_type_def.e_msg_result_def	结果	2	
bet_index	int32	下注索引	3	
bet_count	int32	下注金额	4	

## 续压

说明：在游戏进入到下注阶段时可以调用，请求成功后会自动按照上一盘的下注情况来下注

前端请求协议:packetc2l\_repeat\_bet

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_c2l_repeat_bet 10006

后端返回协议:packetl2c\_repeat\_bet\_result

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_l2c_repeat_bet 15006
result	msg_type_def.e_msg_result_def	结果	2	1.成功 非1.失败
self_bet_list	int64[]	自己下注	3	

## 通知开始押注

说明：服务端通知客户端，表示游戏进入下注状态了，可执行下注和续压请求

后端返回协议:packetl2c\_bc\_begin\_bet

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_l2c_bc_begin_bet 15011
rate_index	int32	赔率ID	2	
colors	int32[]	24个数值 0:红 1:绿 2:黄	3	
cd_time	int32	当前状态剩余时间	4	

## 通知开奖

说明：游戏开始结算，客户端收到消息后，可以开始播放结算相关内容

后端返回协议:packetl2c\_bc\_begin\_award

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_l2c_bc_begin_award 15013
m_award_type	int32	中奖类型	2	
m_award_param_list	int32[]	m_result_type:13~17有效时: 13:1~12编号 list 14:1~12编号,倍数 15:1~12编号,彩金倍数 16:0~3 狮子/熊猫/猴子/兔子 17:0~2 红绿黄	3	
win_gold	int64[]	12个区域	4	
player_reward_list	player_info[]	玩家列表	5	
cd_time	int32	当前状态剩余时间	6	

## 所有玩家下注信息

说明：服务端通知客户端，在游戏下注阶段，服务端会多次向客户端推送此消息

用于同步房间内player的下注情况，只通知每个下注区下了多少

后端返回协议:packetl2c\_bc\_sync\_bets

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_l2c_bc_sync_bets 15012
bet_info_list	int64[]	下注信息	2	

## 游戏控制

说明：调试用消息，只有拥有debug权限的账号才能请求，控制最近一盘开奖结果

前端请求协议:packetc2l\_game\_control

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_c2l_game_control 10022
control	int32	控制类型 1和 2庄 3闲	2	

后端返回协议:packetl2c\_game\_control

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_l2c_game_control 15022
result	msg_type_def.e_msg_result_def	1.成功 非1.失败	2	

## debug信息

说明：调试信息，服务端主动下发房间的监控数据给debug权限的玩家，用于在debug界面查看游戏信息

后端返回协议:packetl2c\_bc\_debuginfo

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_l2c_bc_debuginfo 15023
player_id	int32	玩家ID	2	
total_profit	int64	抽水	3	0
total_stock	int64	库存	4	0
total_Rate	int64	抽水率	5	0
debug_infos	string	调试信息	6	

## 通知所有玩家下注信息

说明：服务端主动下发给有debug权限的玩家，每个真实玩家的下注情况，展示在debug界面

后端返回协议:packetl2c\_notice\_all\_bet\_info

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_l2c_notice_gm_all_bet_info 15021
bet_info_list	gm_msg_bet_info[]	玩家信息链表	2	

# 提醒离开

说明：暂时没有使用该消息

后端返回协议:packet12c\_attention\_needLeave

参数名	类型	描述	参数ID	默认值
packet_id	<a href="#">e_server_msg_type</a>	协议Id	1	e_mst_attention_needLeave 15029

# 协议ID枚举

```
enum e_server_msg_type
{
    //客户端到服务端
    e_mst_start_c2l = 10000;
    e_mst_c2l_leave_room = 10002;//离开房间
    e_mst_c2l_get_room_scene_info = 10003;//获得桌子内信息
    e_mst_c2l_check_state = 10004;//检测游戏状态
    e_mst_c2l_add_bet = 10005;//下注
    e_mst_c2l_repeat_bet = 10006;//重复押注

    e_mst_c2l_game_control = 10022;           //debug调试

    // 服务端到客户端-----
    e_mst_start_l2c = 15000;
    e_mst_l2c_leave_room_result = 15002;//离开房间
    e_mst_l2c_get_room_scene_info = 15003;//获得桌子内信息
    e_mst_l2c_check_state_result = 15004;//检测游戏状态
    e_mst_l2c_add_bet = 15005;//下注
    e_mst_l2c_repeat_bet = 15006;//重复押注

    e_mst_l2c_bc_begin_bet = 15011;          //场景下注状态
    e_mst_l2c_bc_sync_bets = 15012; //场景下注同步
    e_mst_l2c_bc_begin_award = 15013;        //场景开奖状态

    e_mst_l2c_notice_gm_all_bet_info = 15021;           //玩家列表
    e_mst_l2c_game_control = 15022;//游戏控制反馈
    e_mst_l2c_bc_debuginfo = 15023;//调试信息

    e_mst_attention_needLeave = 15029;
    e_mst_clend_index = 20000;
}
```

# 游戏状态



```
enum e_game_state {  
    e_state_game_begin = 0; //游戏开始  
    e_state_game_bet = 1;    //押注期间  
    e_state_game_award = 2;  //开奖期间  
}
```