1. 开房协议

int length = message.readInt32();//数据长度（先传数据长度）

message.readBytes(msg);

CreateRoom createRoom = CreateRoom.Parser.ParseFrom(msg);//proto协议,createroom.proto

1. 房间信息协议

NetworkMessage message = NetworkMessage.Create((ushort)Opcodes.Server\_Room\_Info, 100);

//依照以前数据保留，要限度两个uint8，然后获取长度再读取二进制是聚聚

message.writeUInt8((byte)RoomMessageType.RoomInfo);//房间信息

message.writeUInt8((byte)roomType);//房间类型

message. writeInt64 (memoryStream.Length); //数据长度（必须放进去）

SendRoomInfo sendRoomInfo = new SendRoomInfo();//协议，S2C.proto

1. 发牌协议修改，发牌时就要现实黑桃3，最为找朋友的依据，zhaungposition就是第一个拿到黑桃3的人，也是之后出牌人（红色为修改部分，其他为以前代码，未修改）

NetworkMessage message = NetworkMessage.Create((ushort)Opcodes.Server\_Room\_Info, 100);

message.writeUInt8((byte)RoomMessageType.PlayerHoldCards);//玩家手牌

message.writeUInt32(codeId);//房间号

message.writeUInt8((byte)player\_info.holdCards.Count);//手牌数量

foreach (var item in player\_info.holdCards)

message.writeUInt32(item);//牌

//是否有两张黑桃3用于前端显示，如果有两张那两张显示在同一个人面前，否则两个人一人显示一张

message.writeBool(has\_two);//庄是不是有两张黑桃3

message.writeUInt8((byte)zhuangPosition);//庄的位置(即之后第一个出牌人的位置)

message.writeUInt8((byte)friendPostion);//朋友的位置（另一个黑桃3的位置，或者直接是庄对面）

* ~~此处协议取消~~
* ~~message.writeBool(need\_tp);//是否要摊牌，如果要摊牌，则不能出现出牌按钮，过2秒后服务器~~会发送游戏结束协议

tmp\_player.sendMessage(message);

1. 出牌后协议

NetworkMessage message = NetworkMessage.Create((ushort)Opcodes.Server\_Room\_Info, 100);

message.writeUInt8((byte)RoomMessageType.playerOperate);//玩家操作

message.writeUInt32(codeId);//房间号

message.writeUInt8((byte)player\_info.position);//玩家位置

message.writeUInt8((byte)type);//操作类型

message.writeInt32(operateCard.Count);//出牌的数量

foreach (var item in operateCard) message.writeUInt32(item);

message.writeInt32(player\_info.holdCards.Count);//剩余牌的数量

//if (roomType == RoomType.PK) message.writeUInt32(player\_info.taoShangSocre);//当前淘善分

//else

//{

// message.writeUInt8((byte)drop\_card\_position);//吃 碰 杠 弃牌的玩家

// message.writeBool(isAnGang);//是否是暗杠

//}

message.writeBool(player\_info.cur\_taoShangScore != 0);//当前回合是否有分数变化,有分数变化才读取下面数据

if (player\_info.cur\_taoShangScore != 0) {

message.writeInt32(sitPlayerDict.Count);//玩家人数

foreach (XYRoomPlayerInfo item in sitPlayerDict.Values)

{

message.writeUInt8((byte)item.position);//玩家位置

message.writeInt32(item.cur\_taoShangScore);//当前讨赏变化分数(每个玩家不同，需要播放数据改变动画，飘一下扣的数值)

message.writeInt32(item.taoShangSocre);//当前局讨赏分数（每个人头像下显示的分数）

}

broadcastMessage(message);

1. 游戏结束协议

NetworkMessage message = NetworkMessage.Create((ushort)Opcodes.Server\_Room\_Info, 100);

//以下两个uint8为原代码保留，用于前端识别是哪个房间，定位到具体方法

message.writeUInt8((byte)RoomMessageType.GameOver);//牌局结算

message.writeUInt8((byte)roomType);//房间类型

//以下为定位到咸鱼狗中咸鱼狗棋牌方法后的数据

//以下为识别房间类型后具体的数据，用protobuf转

message.writeInt64(memoryStream.Length);//protobuf长度，

message.writeBytes(memoryStream.ToArray());//S2C中的GameOver

1. 包牌询问（服务器发给客户端）

NetworkMessage message = NetworkMessage.Create((ushort)Opcodes.Server\_Room\_Info, 100);

message.writeUInt8((byte)RoomMessageType.BaoPai);//询问包牌

message.writeUInt32(codeId);//房间号

message.writeUInt8((byte)position);//询问的位置

broadcastMessage(message);

1. 询问包牌返回(客户端发给服务器)

Handler.addPlayerHandler(Opcodes.Client\_Ask\_Bao\_Pai\_Result, onClientBaoPaiResult);//询问包牌返回结果

uint code\_id = message.readUInt32();//房间号

bool isAgree = message.readBool();//是否包牌

1. 广播包牌人位置

NetworkMessage message = NetworkMessage.Create((ushort)Opcodes.Server\_Room\_Info, 100);

message.writeUInt8((byte)RoomMessageType.BaoPaiPosition);//询问包牌

message.writeUInt32(codeId);//房间号

message.writeUInt8((byte)position);//询问的位置

broadcastMessage(message);

九、广播第一个人出牌

//广播第一个人出牌和在打的过程中广播下个玩家出牌是同个协议

//广播第一个人出牌passposition的值为PositionType.None

NetworkMessage message = NetworkMessage.Create((ushort)Opcodes.Server\_Room\_Info, 100);

message.writeUInt8((byte)RoomMessageType.PlayerPass);//玩家过牌

message.writeUInt32(codeId);//房间号

message.writeUInt8((byte)passPosition);//玩家位置

message.writeUInt8((byte)nextPosition);//下一个玩家位置

message.writeUInt8((byte)type);//是否把操作位给朋友

broadcastMessage(message);