**二人麻将文档**

1. 公共宏定义

#define KIND\_ID 507 //游戏 I D

#define GAME\_PLAYER 2 //游戏人数

#define GAME\_NAME TEXT("二人麻将") //游戏名字

#define GAME\_GENRE (GAME\_GENRE\_SCORE|GAME\_GENRE\_MATCH|GAME\_GENRE\_GOLD) //游戏类型

//游戏状态

#define GS\_MJ\_FREE GS\_FREE //空闲状态

#define GS\_MJ\_PLAY (GS\_PLAYING+1) //游戏状态

//常量定义

#define MAX\_WEAVE 4 //最大组合

#define MAX\_INDEX 34 //最大索引

#define MAX\_COUNT 14 //最大数目

#define MAX\_REPERTORY 64 //最大库存

//组合子项

struct CMD\_WeaveItem

{

BYTE cbWeaveKind; //组合类型

BYTE cbCenterCard; //中心扑克

BYTE cbPublicCard; //公开标志

WORD wProvideUser; //供应用户

};

1. 服务器端命令和消息结构

//服务器命令结构

#define SUB\_S\_GAME\_START 100 //游戏开始

#define SUB\_S\_OUT\_CARD 101 //出牌命令

#define SUB\_S\_SEND\_CARD 102 //发送扑克

#define SUB\_S\_LISTEN\_CARD 103 //听牌命令

#define SUB\_S\_OPERATE\_NOTIFY 104 //操作提示

#define SUB\_S\_OPERATE\_RESULT 105 //操作命令

#define SUB\_S\_GAME\_END 106 //游戏结束

#define SUB\_S\_TRUSTEE 107 //用户托管

//游戏状态

struct CMD\_S\_StatusFree

{

LONG lCellScore; //基础金币

WORD wBankerUser; //庄家用户

bool bTrustee[GAME\_PLAYER]; //是否托管

};

//游戏状态

struct CMD\_S\_StatusPlay

{

//游戏变量

LONG lCellScore; //单元积分

WORD wSiceCount; //骰子点数

WORD wBankerUser; //庄家用户

WORD wCurrentUser; //当前用户

//状态变量

BYTE cbActionCard; //动作扑克

BYTE cbActionMask; //动作掩码

BYTE cbHearStatus[GAME\_PLAYER]; //听牌状态

BYTE cbLeftCardCount; //剩余数目

bool bTrustee[GAME\_PLAYER]; //是否托管

//出牌信息

WORD wOutCardUser; //出牌用户

BYTE cbOutCardData; //出牌扑克

BYTE cbDiscardCount[GAME\_PLAYER]; //丢弃数目

BYTE cbDiscardCard[GAME\_PLAYER][60]; //丢弃记录

//扑克数据

BYTE cbCardCount; //扑克数目

BYTE cbCardData[MAX\_COUNT]; //扑克列表

BYTE cbSendCardData; //发送扑克

//组合扑克

BYTE cbWeaveCount[GAME\_PLAYER]; //组合数目

CMD\_WeaveItem WeaveItemArray[GAME\_PLAYER][MAX\_WEAVE]; //组合扑克

};

//游戏开始

struct CMD\_S\_GameStart

{

WORD wSiceCount; //骰子点数

WORD wBankerUser; //庄家用户

WORD wCurrentUser; //当前用户

BYTE cbUserAction; //用户动作

BYTE cbCardData[MAX\_COUNT]; //扑克列表

bool bTrustee[GAME\_PLAYER]; //是否托管

};

//出牌命令

struct CMD\_S\_OutCard

{

WORD wOutCardUser; //出牌用户

BYTE cbOutCardData; //出牌扑克

};

//发送扑克

struct CMD\_S\_SendCard

{

BYTE cbCardData; //扑克数据

BYTE cbActionMask; //动作掩码

WORD wCurrentUser; //当前用户

};

//听牌命令

struct CMD\_S\_ListenCard

{

WORD wListenUser; //听牌用户

};

//操作提示

struct CMD\_S\_OperateNotify

{

WORD wResumeUser; //还原用户

BYTE cbActionMask; //动作掩码

BYTE cbActionCard; //动作扑克

};

//操作命令

struct CMD\_S\_OperateResult

{

WORD wOperateUser; //操作用户

WORD wProvideUser; //供应用户

BYTE cbOperateCode; //操作代码

BYTE cbOperateCard; //操作扑克

};

//游戏结束

struct CMD\_S\_GameEnd

{

LONG lGameTax; //游戏税收

//结束信息

WORD wProvideUser; //供应用户

BYTE cbProvideCard; //供应扑克

DWORD dwChiHuKind[GAME\_PLAYER]; //胡牌类型

DWORD dwChiHuRight[GAME\_PLAYER]; //胡牌类型

//积分信息

LONG lGameScore[GAME\_PLAYER]; //游戏积分

//扑克信息

BYTE cbCardCount[GAME\_PLAYER]; //扑克数目

BYTE cbCardData[GAME\_PLAYER][MAX\_COUNT]; //扑克数据

};

//用户托管

struct CMD\_S\_Trustee

{

bool bTrustee; //是否托管

WORD wChairID; //托管用户

};

1. 客户端命令和消息结构

//客户端命令结构

#define SUB\_C\_OUT\_CARD 1 //出牌命令

#define SUB\_C\_LISTEN\_CARD 2 //听牌命令

#define SUB\_C\_OPERATE\_CARD 3 //操作扑克

#define SUB\_C\_TRUSTEE 4 //用户托管

//出牌命令

struct CMD\_C\_OutCard

{

BYTE cbCardData; //扑克数据

};

//操作命令

struct CMD\_C\_OperateCard

{

BYTE cbOperateCode; //操作代码

BYTE cbOperateCard; //操作扑克

};

//用户托管

struct CMD\_C\_Trustee

{

bool bTrustee; //是否托管

};

1. 框架类接口

//初始函数

virtual bool InitGameFrame();

//重置框架

virtual void ResetGameFrame();

//游戏设置

virtual void OnGameOptionSet();

//时间消息

virtual bool OnTimerMessage(WORD wChairID, UINT nElapse, UINT nTimerID);

//旁观状态

virtual void OnLookonChanged(bool bLookonUser, const void \* pBuffer, WORD wDataSize);

//网络消息

virtual bool OnGameMessage(WORD wSubCmdID, const void \* pBuffer, WORD wDataSize);

//游戏场景

virtual bool OnGameSceneMessage(BYTE cbGameStatus, bool bLookonOther, const void \* pBuffer, WORD wDataSize);

1. 视图类接口

//消息定义

#define IDM\_START (WM\_USER+100) //开始消息

#define IDM\_OUT\_CARD (WM\_USER+101) //出牌消息

#define IDM\_TRUSTEE\_CONTROL (WM\_USER+102) //托管控制

#define IDI\_DISC\_EFFECT 102 //丢弃效果

//重置界面

virtual void ResetGameView();

//调整控件

virtual void RectifyGameView(int nWidth, int nHeight);

//绘画界面

virtual void DrawGameView(CDC \* pDC, int nWidth, int nHeight);

1. 逻辑类定义和数据结构

//逻辑掩码

#define MASK\_COLOR 0xF0 //花色掩码

#define MASK\_VALUE 0x0F //数值掩码

//动作定义

//动作标志

#define WIK\_NULL 0x00 //没有类型

#define WIK\_LEFT 0x01 //左吃类型

#define WIK\_CENTER 0x02 //中吃类型

#define WIK\_RIGHT 0x04 //右吃类型

#define WIK\_PENG 0x08 //碰牌类型

#define WIK\_GANG 0x10 //杠牌类型

#define WIK\_LISTEN 0x20 //听牌类型

#define WIK\_CHI\_HU 0x40 //吃胡类型

//胡牌定义(暂时定义)

//牌型掩码

#define CHK\_MASK\_SMALL 0x0000FFFF //小胡掩码

#define CHK\_MASK\_GREAT 0xFFFF0000 //大胡掩码

//小胡牌型

#define CHK\_NULL 0x00000000 //非胡类型

#define CHK\_JI\_HU 0x00000001 //鸡胡类型

#define CHK\_PING\_HU 0x00000002 //平胡类型

//大胡牌型

#define CHK\_PENG\_PENG 0x00010000 //碰碰胡牌

#define CHK\_QI\_XIAO\_DUI 0x00020000 //七小对牌

#define CHK\_SHI\_SAN\_YAO 0x00040000 //十三幺牌

//胡牌权位(暂时定义)

//牌权掩码

#define CHR\_MASK\_SMALL 0x000000FF //小胡掩码

#define CHR\_MASK\_GREAT 0xFFFFFF00 //大胡掩码

//大胡权位

#define CHR\_DI 0x00000100 //地胡权位

#define CHR\_TIAN 0x00000200 //天胡权位

#define CHR\_QING\_YI\_SE 0x00000400 //清一色牌

#define CHR\_QIANG\_GANG 0x00000800 //抢杆权位

#define CHK\_QUAN\_QIU\_REN 0x00001000 //全求权位

//类型子项

struct tagKindItem

{

BYTE cbWeaveKind; //组合类型

BYTE cbCenterCard; //中心扑克

BYTE cbCardIndex[3]; //扑克索引

};

//组合子项

struct tagWeaveItem

{

BYTE cbWeaveKind; //组合类型

BYTE cbCenterCard; //中心扑克

BYTE cbPublicCard; //公开标志

WORD wProvideUser; //供应用户

};

//胡牌结果

struct tagChiHuResult

{

DWORD dwChiHuKind; //吃胡类型

DWORD dwChiHuRight; //胡牌权位

DWORD dwWinTimes; //番数数目

};

//杠牌结果

struct tagGangCardResult

{

BYTE cbCardCount; //扑克数目

BYTE cbCardData[4]; //扑克数据

BYTE cbGangType; //杠牌类型

};

//分析子项

struct tagAnalyseItem

{

BYTE cbCardEye; //牌眼扑克

BYTE cbWeaveKind[4]; //组合类型

BYTE cbCenterCard[4]; //中心扑克

};

//扑克数据

const BYTE CGameLogic::m\_cbCardDataArray[MAX\_REPERTORY]=

{

0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09, //万子

0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09, //万子

0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09, //万子

0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09, //万子

//0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19, //索子

//0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19, //索子

//0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19, //索子

//0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19, //索子

//0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x29, //同子

//0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x29, //同子

//0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x29, //同子

//0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x29, //同子

0x31,0x32,0x33,0x34,0x35,0x36,0x37, //番子

0x31,0x32,0x33,0x34,0x35,0x36,0x37, //番子

0x31,0x32,0x33,0x34,0x35,0x36,0x37, //番子

0x31,0x32,0x33,0x34,0x35,0x36,0x37, //番子

};

7.扑克控制

//扑克定义

#define HEAP\_FULL\_COUNT 32 //堆立全牌

//X 排列方式

enum enXCollocateMode

{

enXLeft, //左对齐

enXCenter, //中对齐

enXRight, //右对齐

};

//Y 排列方式

enum enYCollocateMode

{

enYTop, //上对齐

enYCenter, //中对齐

enYBottom, //下对齐

};

//方向枚举

enum enDirection

{

Direction\_East, //东向

Direction\_South, //南向

Direction\_West, //西向

Direction\_North, //北向

};