接口说明

* 整个项目分成四个工程
* DdzAiServer，这个是运行ai的服务器进程，这个进程会开启16个线程进行斗地主ai计算。
* DdzAiPublic 这个是接口与斗地主ai服务器进程，进行socket通讯的消息封包的数据结构
* DdzAiLib，这个是内嵌入服务器，与斗地方进行通讯的库项目。
* DdzAiSample，这个是一个示例程序，用于展示，如何使用ai lib项目。
* ai服务器和ai lib库进行通讯的，服务器配置。
* 在DdzAiServer和DdzAiSample下，有一个配置文件，communication.cfg ,这个用来配置斗地主ai和斗地主库进行 socket通讯的文件。
* 配置如下 斗地主ai服务器的配置为

[Communication2]

LocalIP=127.0.0.1 服务器的ip

LocalPort=4001 服务器的端口

ConnectCount=1 有多少个前台服务连接到本服务器

[Connect\_1]

ConnectIP=127.0.0.1 连接到本服务器的ip

ConnectPort=2000 连接到本服务器的端口

ConnectServerType=1 连接到本服务器的服务器类型 1为前台服务 2为后台服务

ConnectServerId=2 连接到本服务器的服务器ip

一般配置为，先启动后台服务器，然后等待前台服务器进行连接。

连接到斗地主服务器的库通讯配置

[Communication2]

LocalIP=127.0.0.1 前台服务器ip

LocalPort=2000 前台服务器端口

ConnectCount=1 需要连接后台服务器的数量

[Connect\_1]

ConnectIP=127.0.0.1 后台服务器ip

ConnectPort=4001 后台服务器端口

ConnectServerType=2 后台服务器类型

ConnectServerId=1 后台服务器id

* 斗地主接口数据结构说明，在ai中，不区别桃信梅方的牌类型
* 牌定义

pai\_type\_none = 0,

pai\_type\_3 = 3,

pai\_type\_4 = 4,

pai\_type\_5 = 5,

pai\_type\_6 = 6,

pai\_type\_7 = 7,

pai\_type\_8 = 8,

pai\_type\_9 = 9,

pai\_type\_10 = 10,

pai\_type\_J = 11,

pai\_type\_Q = 12,

pai\_type\_K = 13,

pai\_type\_A = 14,

pai\_type\_2 = 15,

pai\_type\_blackjack = 16, // 小王

pai\_type\_blossom = 17, // ´大王

pai\_type\_max = 18,

传给接口的每家的牌，是一个unsigned char pai\_list[pai\_type\_max] 其中下标为牌定义，数组内容为此类牌有几张。

2、出牌数据结构定义

Ai算出需要出什么样的牌，由出牌数据结构给出。

struct pai\_interface\_move

{

unsigned char \_type; 这个是ai的出牌类型，见下表

unsigned char \_alone\_1;

unsigned char \_alone\_2;

unsigned char \_alone\_3;

unsigned char \_alone\_4;

unsigned char \_airplane\_pairs;

unsigned char \_combo\_list[20];

unsigned char \_combo\_count;

};

其中\_type为出牌的类型，其他数据为些类型所带的参数，见下表

1、 ddz\_type\_no\_move = 0,

这个数据类型为，ai决定此轮，将不出牌。

2、 ddz\_type\_alone\_1 = 1,

这个数据类型为，ai决定此轮，出一张单牌，其中\_alone\_1为这张单牌的牌值。

3、ddz\_type\_pair = 2,

这个数据类型为，ai 决定此轮，出一对牌，其中\_alone\_1为这张对牌的牌值。

4、ddz\_type\_triple = 3,

这个数据类型为，ai决定此轮，出一个三张的牌，不带一也不带一对，其中\_alone\_1为这三张牌的牌值

5、 ddz\_type\_triple\_1 = 4,

这个数据类型为，ai决定此轮，出一个三带一的牌型，其中\_alone\_1 为三张的牌值，\_alone\_2为带一张的牌值

6、ddz\_type\_triple\_2 = 5,

这个数据类型为，ai决定此轮，出一个三带二的牌型，其中\_alone\_1 为三张的牌值，\_alone\_2为带一对的牌值

7、ddz\_type\_order = 6,

这个数据类型为，ai决定此轮出一个顺子牌，其中\_alone\_1 \_alone\_2 \_alone\_3 \_alone\_4都为0，顺子的牌存储在\_combo\_list 这个数组中，其中\_combo\_count为这个顺子牌的个数

例如 345678

8、ddz\_type\_order\_pair = 7,

这个数据类型为，一个连对的牌，其中\_alone\_1 \_alone\_2 \_alone\_3 \_alone\_4都为0，连对的牌存储在\_combo\_list 这个数组中，其中\_combo\_count为这个连对牌的个数

例如 33445566

9、ddz\_type\_airplane = 8,

这个数据为，飞机不带牌的牌型，其中\_alone\_1 \_alone\_2 \_alone\_3 \_alone\_4都为0，飞机的牌存储在\_combo\_list 这个数组中，其中\_combo\_count为这个飞机牌的个数

例如 555666

10、ddz\_type\_airplane\_with\_pai = 9,

这个数据为，飞机带牌的牌型，其中\_alone\_1 \_alone\_2 \_alone\_3 \_alone\_4都为0，飞机的牌存储在\_combo\_list 这个数组中，其中\_combo\_count为这个飞机牌的个数

例如 555666

带牌在\_alone\_1 \_alone\_2 \_alone\_3 \_alone\_4中，飞机最多带四张单牌，和四个对子

如果带四个对子，则\_airplane\_pairs为真

例如： \_alone\_1 == 3 \_alone\_2 == 4 airplane\_pairs == true 则带一对3和一对4,

\_alone\_1 == 3 \_alone\_2 == 3 airplane\_pairs == false 则带二张单牌，为二个3

11、ddz\_type\_bomb = 10,

这个牌型为四张的炸弹。\_alone\_1 为炸弹的牌值

12、ddz\_type\_king\_bomb = 11,

这个牌型为王炸，参数都为零

13、ddz\_type\_i\_four\_with\_alone1 = 12,

这个牌型为四带二，参数都为零

\_alone\_1 为炸弹牌值，\_alone\_2散牌1，\_alone\_3散牌2

13、ddz\_type\_i\_four\_with\_pairs = 13,

这个牌型为四带二，参数都为零

\_alone\_1 为炸弹牌值，\_alone\_2对牌1，\_alone\_3对牌2

* 项目编译顺序
* 首先编译，DdzAiPublic 目录下，src的子目录中的项目
* 再编译 DdzAi这个目录下的，斗地主ai服务器
* 再编译 DdzAiLIb这个目录下的，连接斗地主ai服务器的库项目
* 最后编译 DdzAiSample这个示例项目。
* 斗地主的接口库文件，主要的接口库文件为DdzAiInterface.h这个文件中，接口设计为异步方式，每一个接口调用，不马上返回值，将从回调中返回具体的值。
* CreateAI 接口，这个接口主要是产生一个调用序列值，后面的回调将返回这个值作为调口异步调用的凭证。
* AiPlayCard 这个是ai主动出牌接口，这个接口会传三个玩家的手中牌给ai，其中要出牌的玩家，放在第一个位置，后面二个玩家按出牌顺序放置，然后，还要标明哪个玩家是地主。Ai将根据手牌给出要出牌的序列。
* AiPlayCardPassive 这个是ai被动出牌接口，当牌桌上有需要ai管住的牌时调用，这个接口也需要传三个玩家的手牌信息，和谁是地主信息。然后这个接口比主动出牌接口多了一个牌桌上牌的数据。这个数据按文档的牌型数据结构来填充
* GrabLandlord 这个是抢地主接口，ai抢地主叫多少分，在这个接口里设置。
* SetPlayCardInterface 这个是设置异步回调的接口。
* 需要在服务器进程的循环里调用接口的Update函数，用于把ai服务器传过来的封包更新给服务器进程。
* 增加一个GrabLandlord2的接口，这个接口会实时运算一下，返回一个抢地主能否打赢的0-100概率值，现在只运算10次。