**主要程序说明：**

B 服务器 （中心服务器）-- BZGameGate.bcf 主要功能为将基础配置发到客户端，通知客户端Z服务器地址及其它配置

Z服务器 （登入服务器）-BZGameLogon.BCF 主要功能为负责玩家登入平台，及道俱，玩家在大厅内玩家操作回应 （需要建立数据库联接，并操作数据库）

W服务器 (游戏服务器) - BZGameLocal.BCF 服务游戏房间内操作。（需要建立数据库联接，并操作数据库）加载每个游戏DLL

大厅.exe 登入玩家进入平台，并启动GameLauncher.exe

GameLauncher.exe 调用游戏ico，与大厅使用管道通信。GameLauncher启动游戏现在分为两种即百宝湾编辑器开发的游戏和GDI开发的游戏。

**平台启动运行流程：**

玩家启动大厅，程序读取bzgame.bcf B服务器地址，与B建立联接，建立联接后，B服务器将Z服务器地址，端口及其它配置发到平台客户，客户端接收后，断开与B联接。创建MainRoom，与Z建立联接，弹出登入框。输入登入信息，点确定，将密码加密，发送到Z服务器，服务器接收后，压入数据库处理队列，调用数据存诸过程判定登入合法。数据库处理过程完毕后，依结果向客户端返回，正确登入Z记录用户Id及Socket 索引。Z写数据库标记玩家登入。客户端收到登入成功后界面表现，并向服务器发送请求游戏列表请求。Z服务器发送游戏列表。客户端接收，接收完成显示，隐藏正在登入提示框。当右边游戏列表被占击时，向服务器发送请求房间列表。

点击进入房间，建立GameRoom，GameRoom建立与W联接Socket。联接成功，再将玩家信息，登入密码发送到W。压入数据库处理队列处理玩家登入请求（存储过程执行），存储过程会对密码，是否重复登入等进入检测。返回成功后，发送房间信息到玩家客户端。客户端表现。

玩家点击桌子坐到坐子上。向W服务器发送坐下消息，服务器返回，客户端建立GameFrame，GameFrame建立IPC管道通信;

GameFrame 启动GameLauncher进程。GameLauncher程序启动后建立IPC ;IPC联通后，GameLauncher通过要加载的游戏是否百宝湾游戏来确立不同的初始化游戏；加载游戏后，游戏与服务器的消息全部都是通过ipc发到平台再由平台通过socket发到服务端，消息接收一样；

**游戏运行游戏程：**

玩家点击坐位，大厅向服务器发坐下消息，服务器验证返回,服务器玩家加入桌子玩家列表，大厅启动GameLauncher，GameLauncher，加载游戏，并向服务器发送客户端自已信息（ASS\_GM\_GAME\_INFO），是否允许旁观。服务端接收后设置游戏状态，调用游戏OnGetGameStation；正式进入游戏流程。通过OnGetGameStation发送游戏状态到玩家客户端，客户端SetGameStation接收并设置游戏场景，若是未开始游戏，玩家点击开始按钮，游戏向服务器发送举手消息。服务器接收举手消息，依条件判定是否可以开始游戏，达到开始条件，调用GameBegin虚函数。游戏执行游戏服务器的GameBegin，游戏开始，进入玩游戏流程

**游戏框架：**

**服务端：**

游戏逻辑类（即桌子，继承服务端GameDesk类）负责游戏的流程走向，与客户端消息交互）

重要虚函数：

GameBegin(BYTE bBeginFlag) 函数 游戏开始函数，一张桌子开始游戏，此函数为游戏入口，为平台调用

GameFinish函数 游戏结束处理函数，一般结算在此函数中进行，游戏自已调用

HandleNotifyMessage函数 游戏消息处理函数

OnGetGameStation函数 玩家进入游戏向客户端置场景功能，包括游戏没开始或者断线重回都从要进入这里，然后 消息到

IsPlayGame 判定当前是否在游戏中，可在游戏里重新实现，

UserLeftDesk 玩家离开桌子回调函数,

UserNetCut 玩家断线回调函数

**客户端**：

游戏逻辑（即现有GameClientDlg，）负责客户端游戏流程处理，服务端消息接收，玩家UI操作处理发送

视图（GDI游戏有，百宝湾游戏无此类,百宝湾与游戏逻辑在一块）负责界面显示

SetGameStation函数 设置场景函数，相对应服务端OnGetGameStation的消息处理

HandleGameMessage函数 服务端消息处理函数

GameUserLeft 玩家退出

GameUserCome 玩家进入

**其它：**

GameMessage 公共消息定义，数据包协议定义，

游戏规则类，（牌型判定，是否可以出牌等一系列逻辑算法），一般可以与客户端共用