# 红鸟平台开发指南



文档名称

红鸟平台客户端开发指南

红鸟网络科技有限公司

日期

2018-3-14

开发部

针对平台

红鸟平台

版权

红鸟网络科技

目录

[一、概述 **4**](#_TOC_250049)

[2、平台客户端结构组成 **5**](#_TOC_250048)

* 1. [1什么是大厅 5](#_TOC_250047)
  2. [、大厅组件 7](#_TOC_250046)
  3. [、什么是房间 7](#_TOC_250045)
  4. [、房间的细节 8](#_TOC_250044)
  5. 、游戏的基本 GameLauncher 10
  6. [、游戏 10](#_TOC_250043)
     1. [游戏的分类 10](#_TOC_250042)
     2. [游戏的安装路径与启动方法 11](#_TOC_250041)
  7. [更新机制 12](#_TOC_250040)

[大厅的更新 13](#_TOC_250039)

[游戏的更新 14](#_TOC_250038)

[二.客户端程序设计思路 **15**](#_TOC_250037)

* 1. [1大厅设计概述 15](#_TOC_250036)
  2. [2大厅进程类 CGamePlaceApp与大厅主窗口类 CGamePlaceDlg 15](#_TOC_250035)
  3. [3大厅顶部窗口 CTopDuWndClass 15](#_TOC_250034)
  4. [4大厅游戏列表窗口 CGameListWnd 16](#_TOC_250033)
  5. [5大厅主程类 CMainRoomEx 16](#_TOC_250032)
  6. [6房间设计概述 18](#_TOC_250031)
  7. [7房间的右则窗口 CRoomRightWnd 18](#_TOC_250030)
  8. [8房间桌子 CDeskFrameView 19](#_TOC_250029)
  9. 房间对游戏的控制类 CLoveSendClassForEXE. 20
  10. 游戏的启动 21

[三.客户端与服务器的关系 **22**](#_TOC_250028)

* 1. [1 红鸟服务器群落 22](#_TOC_250027)
  2. [2 客户端与 A 服务器 23](#_TOC_250026)
  3. [3 客户端与 M 服务器 24](#_TOC_250025)
  4. [4 游戏房间与 G 服务器 25](#_TOC_250024)
  5. [5 红鸟的网络消息包的结构 26](#_TOC_250023)

[四.房间与游戏进程的关系 **27**](#_TOC_250022)

* 1. [IFrameInterface接口 27](#_TOC_250021)
  2. [大厅的房间与游戏两进程的底层通讯方式 28](#_TOC_250020)
  3. [3房间与游戏几个重要的 IPC消息 29](#_TOC_250019)

[五.红鸟产品的界面技术 **30**](#_TOC_250018)

[5. 1 红鸟主流界面开发技术概述 30](#_TOC_250017)

[六.平台服务端结构组成概述 **31**](#_TOC_250016)

[6.1 服务端主要组成模块及其功能 31](#_TOC_250015)

[七.平台服务端程序设计思路 **32**](#_TOC_250014)

* 1. [服务端基本框架 32](#_TOC_250013)
  2. 网络模块：CTCPSocketManage. 33
  3. 逻辑模块基类：CBaseMainManage. 33
  4. [4数据库模块基类：CAFCDataBaseManage 33](#_TOC_250012)
  5. [Aserver框架 34](#_TOC_250011)
  6. [MServer 框架 35](#_TOC_250010)
  7. [GameManage 框架 35](#_TOC_250009)
  8. [8服务端的基本数据流程 36](#_TOC_250008)

[八. GServer与游戏服务端的调用 **38**](#_TOC_250007)

* 1. CGameDesk 提供给游戏服务端的接口 38
  2. [GServer装载游戏服务端 dll的方法 39](#_TOC_250006)

[九.服务端相关的数据库 **39**](#_TOC_250005)

* 1. [AServer服务端使用的主要表 39](#_TOC_250004)
  2. [AServer服务端使用的主要存储过程 39](#_TOC_250003)
  3. [GServer服务端使用的主要表 40](#_TOC_250002)
  4. [GServer服务端使用的主要存储过程 40](#_TOC_250001)
  5. GameManage服务端使用的主要表 40
  6. [GameManage服务端使用的主要存储过程 40](#_TOC_250000)

# 一、概述

红鸟客户端是使用目前网络棋牌游戏最典型的设计，也是公司的经典之作。玩家通过游戏列表进入房间然后坐桌进行游戏，操作直观方便。

红鸟平台与当前市面大多数棋牌平台比较存在的优势有：

1. 界面美化，使用专业的 U I 引擎重构界面。
2. 增强玩家游戏币流通功能，如银行，即时通讯等。
3. 赛制
4. 增加游戏钱包，银行系统再次升级。
5. 开发环境为当前比较新颖的 VS2010；
6. 支持 Unity3D游戏；

图:红鸟 2.0 客户端

# 2、平台客户端结构组成

### 2. 1 什么是大厅

大厅是用户登陆后所看到的界面，是游戏支持游戏运行的程序。大厅还包括用户信息管理，银行，聊天等非游戏组件，不包括游戏。为对大厅的认识可以简单地分为三个部分：顶部导航窗口、游戏列表窗口与房间窗口。这三个部分是大厅的主体，每个部分当中还有许多的细节，在本文档者有介绍。



图:红鸟大厅结构

顶部导航窗口，顶部导航分从左到右分为三个区域：大厅 logo，导航按钮

与设置按钮。其中导航按钮是导航窗口的关键，大厅部分重要的组件由这些按钮启

动，如银行，意见反馈箱等；其二，有些导航按钮可以打开 web页为用户提供一些服务，如个人中心，查看比赛排名，充值等。

游戏列表窗口，游戏列表窗口包括两个子窗口：用户信息面板窗口与游戏列表

控件窗口，两者呈上下排列。



信息面板窗口显示用户的头像，用户名，面板下方有一排按钮，与导航窗口的按钮一样负责启动一些大厅组件，与导航窗口的按钮所负责的组件不同是，信息面板组件功能更偏重绑定已登陆用户服务如活动通知，个人信息管理，I M 等，导航窗口的按钮大多数与用户无关，如进入官方网站，充值，意见反馈箱等。

游戏列表控件，游戏列表是为用户进入房间，查看房间状态(拥挤程度，进入条件等)的控件，游戏列表的总体结构是：游戏类型，游戏名，房间名;



游戏房间窗口，房间窗口是使用 Tab页的方式来展示每一个房间，游戏房间

窗口的第一个 Tab页是用来显示网页，如图 1.2。当用户点击游戏列表进入游戏

房间，游戏房间窗口会添加一个 Tab页，这个 Tab页就是该游戏的房间，里面有

桌子，玩家等。房间的 Tab可以关闭，但排在最前的用于显示网页的 Tab页是不能关闭的。



图:房间的 Tab页

### 、大厅组件

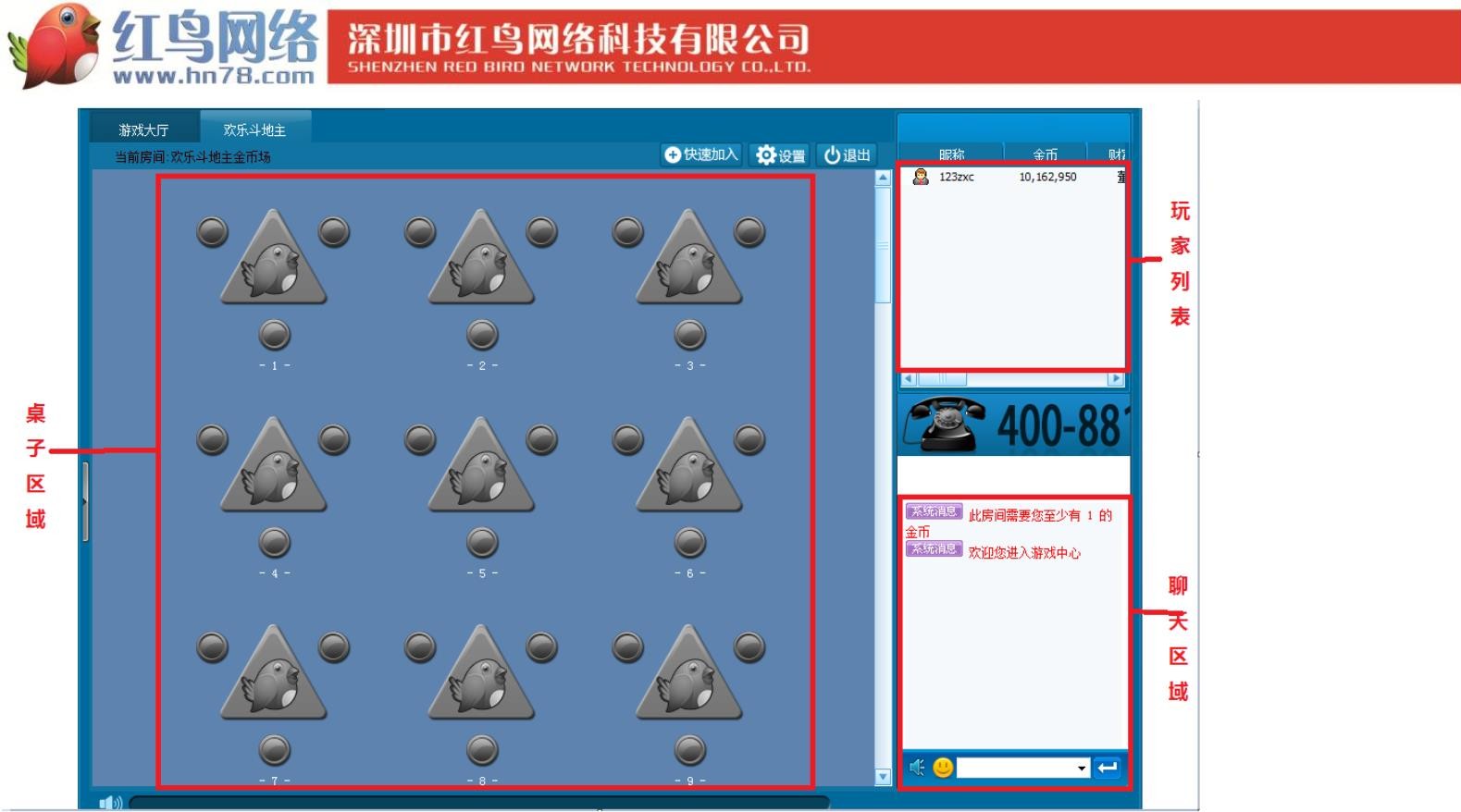
为丰富红鸟平台的功能，价值与体验，特别开发的程序。诸如好友系统，银行，邮箱，魅力值兑换，用户资料管理等都是附属于大厅的组件。大厅的组件有主要特点是组件存与否不影响游戏的运行，不影响大厅的工作秩序。

### 、什么是房间

用户依次点击房间列表的类别，游戏名，双击房间名(如牌类，斗地主 3 人， 双击斗地主 3 人金币房间)，双击后游戏大厅布局出现了调整：游戏列表隐藏； 游戏房间窗口新增一 Tab页，并自动跳转到该 Tab页中，Tab页显示桌子；大厅右则出现了一个玩家列表与聊天框上下并排。Tab页、大厅右则的玩家列表与聊天框形成有有机整体称之为房间。如图 1.6。

T a b 页负责形象地显示房间的状态，桌子数量，玩家状态等等。Tab页另一个重要的任务是提供 U I 令玩家方便地坐桌离桌。

玩家列表是房间中玩家的状态的列表描述，可以方便了解一些特殊资料，如哪个游戏金币最多等。

聊天框，聊天消息同房间的所有玩家会接收到信息，另外房间聊天框也接收小喇叭与大喇叭，GameManager(GM)及系统公告的消息。

图：房间结构

### 、房间的细节

1.房间的工具条，房间工具条有三个功能按钮，分别作用是快速坐桌，房间 设置与退出房间。

快速坐桌功能能为玩家找到最容易开始游戏的桌子，然后坐下，再房间繁忙时该功能十分有用。

房间设置，是这红鸟平台的一个特色，玩家可以制定房间的个性配置，如下图。设置是以房间为单位，不同房间可以有不同的配置。

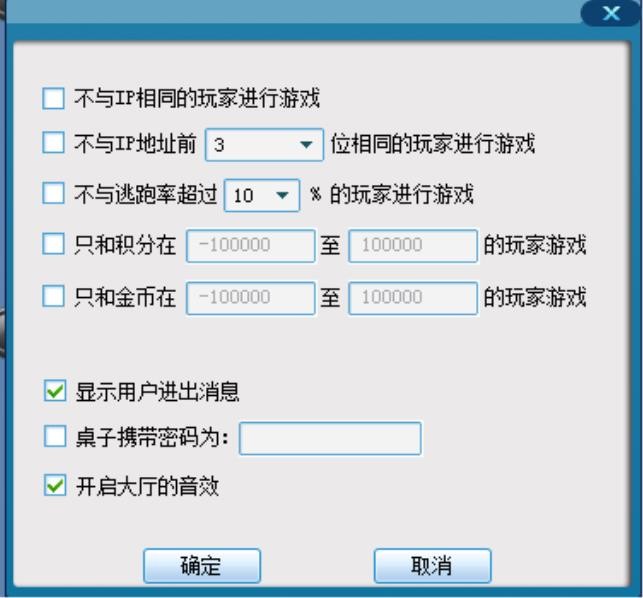


图 1.6房间设置

退出房间，大游戏列表将会重新显示，游戏房间窗口的被退出的房间 Tab 页同删除，房间玩家列表与聊天框也一并关闭消失。此外还可以通过双击当前的

Tab 页来关闭房间。

2.玩家信息查看，右键房间玩家列表玩家或在 Tab 页桌子中右击玩家头像，在弹出的菜单中选择详细信息，即可以查看玩家的信息。这两种方法都是打开同一个信息窗口，只是 UI 操作上有所差别。

3.房间中使用的大厅组件：

银行，银行组件可在大厅使用，如果未开始游戏，在房间内也可使用，如果开始游戏后，银行组件则无法使用。

好友系统，玩家可以通过玩家列表添加好友，发起聊天。但能否可以发起房间的聊天，这与房间的规则配置有关。

4.房间的类型

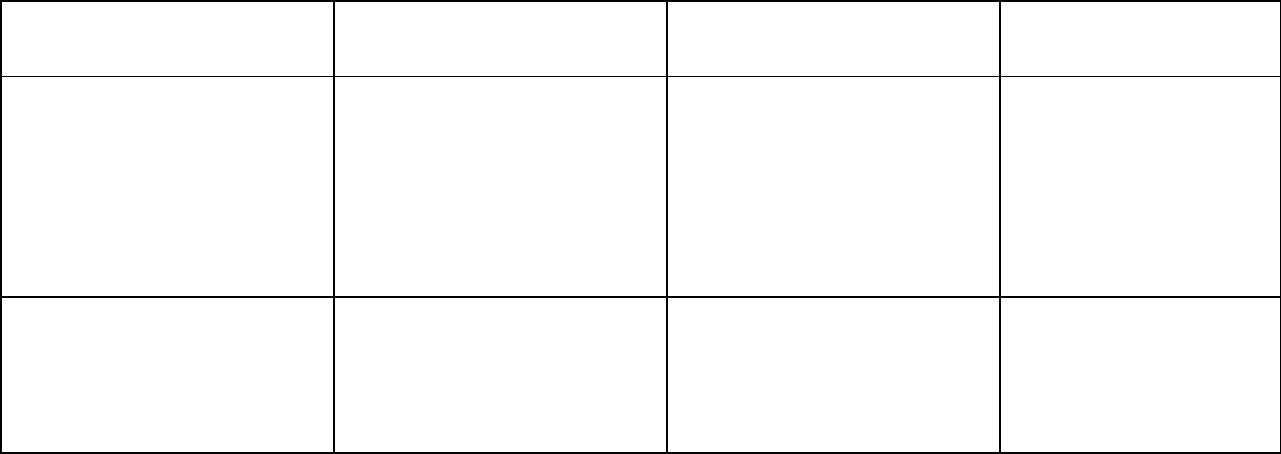
按游戏得失区分，分别使用金币房间，使用积分房间，与练习房间(不消耗 金币与积分);

按游戏游戏人数区分，分别有百家乐房间(即有多 Tab 页只有一个桌子的房间)，与非百人游戏房间(即有多 Tab 页有多桌子的房间);

按制度区分，分别有休闲房间，排队房间，及比赛房间。

这三种房间类型的区分可以交错使用，百家乐练习房间，但也有不能交错的时间，如斗地主的休闲房间不能配置为百家乐房间。所有的房间搭配关系可以用下表来体现:

表 1.1 房间性质搭配关系



百人游戏

休闲类**(**多桌**)**

可配金币房间

排队类**(**单桌**)**

(不存在)

比赛类

(不存在)

可配积分房间

可配练习房间

非百人游戏

可配金币房间

可配积分房间

可配金币房间

可配积分房间

只能配置积分

房间



可配练习房间

可配练习房间

* 1. 、游戏的基本 **GameLauncher**

红鸟平台客户端有两个重要的进程，第一是大厅进程，有于处理用户登陆，提供一切组件，及房间桌子等操作；第二是游戏进程。这里重点介绍游戏进程，游戏的进程就是 GameLauncher.exe，称这个可执行文件为游戏启动器

(GameLauncher)。GameLauncher 却不是游戏的全部，它是游戏的基本。一个完整的游戏程序客户端，除了 GameLauncher 这个基础外，还有游戏逻辑。游戏逻辑程序以 DLL 的形式存在(红鸟游戏逻辑的\*.dll 命名为\*.ico)。大厅启动游戏的过程是，创建 GameLauncher 子进程，GameLauncher 进程加载对应游戏逻辑 DLL 库文件。不同的游戏，逻辑 DLL 库文件的名字、内容不相同，但使用的

GameLauncher 是一样的。所以 GameLauncher 肩负着两个重要的使命：

1.根据大厅要求查找游戏逻辑 DLL 库文件，加载游戏逻辑 DLL 库文件，形成一个游戏特例，之所以能分出斗地主，分出象棋，就是靠这功能；

2.GameLauncher 作为游戏的基本，实现游戏的共性功能。游戏千差万别， 也有一些相似功能模块，如窗口框架，道具，消息通讯等，这些部分由

GameLauncher 实现，游戏开发人员写游戏逻辑 DLL 库文件时就不需要理会这些

细节了。

### 、游戏

游戏与平台各占半边天，平台的价值最终体现在游戏上。

###### 游戏的分类

红鸟游戏的分类有两种方式

* + - 1. 按游戏性质划分，有分扑克类游戏、麻将类游戏、休闲类游戏。
      2. 按游戏聚众方式划分，有百人类游戏与非百人类游戏。

百人类游戏的特点是房间只有一张桌子，最多同时容纳 180位玩家同桌游

戏，如百家乐，生肖转盘，猪跑跑等；非百人类游戏的房间桌子数量自定义，如

斗地主、大众麻将等。

* + - 1. 按游戏实现技术划分，有 MFC游戏、百宝湾 U I 擎游游戏、Unity3D引擎游戏等。

MFC游戏优点是游戏文件体积小，缺点是开发周期长，体验感差。如象棋，

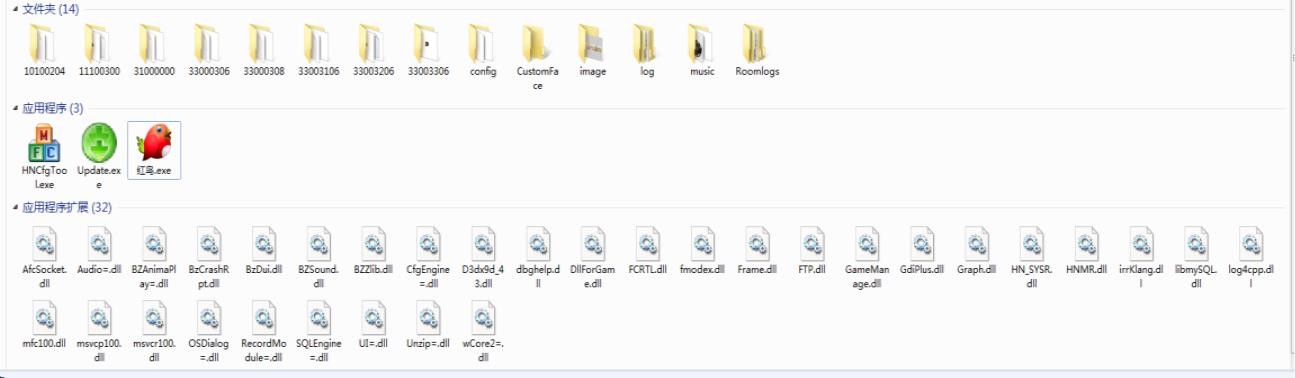
军棋，2 人牛牛等，MFC游戏使用的技术现已被公司所淘汰；

百宝湾 U I 游戏优点是游戏体验好、擅长制作复杂的游戏；缺点是对显存有

要求。如欢乐斗地主，扎金花，德洲扑克等，百宝湾 U I 目前是公司的主流游戏开发技术；

Unity3D引擎游优点是优越的戏游戏体验好、可以实现 3 D 游戏效果，缺点是开发周长，文件体积大。如 3 D 森林舞会。

###### 游戏的安装路径与启动方法

安装路径在大厅的根目录下，每个游戏存放在以其游戏 I D 为名的文件夹中，也称游戏目录。如下图

大厅根目录与游戏目录

红鸟的游戏游戏目录中主要包含：游戏逻辑 DLL库文件(DLL库文件扩展名被修改成 ico)、GameLauncher(GameLauncher.exe与 DllForGame.dll)、游戏的图片声音资源、配置文件(GameOption.bcf、[游戏 ID].bcf)、及第三方库文件等。逻辑 DLL库文件与 GameLauncher是必要的。正如 1.6节所讲，玩家进入在大厅房间坐下后，大厅通过当场的游戏 I D 找到对应游戏目录的

GameLauncher.exe，启动之。启动 GameLauncher.exe时大厅把游戏的参数传入

GameLauncher.exe，这些参数有游戏 ID。GameLauncher.exe利用大厅传入的游

戏 ID，动态加载游戏逻辑库。经过一系列的 U I 初始化，通讯连接，游戏界面就出现，完成游戏的启动。

### 更新机制

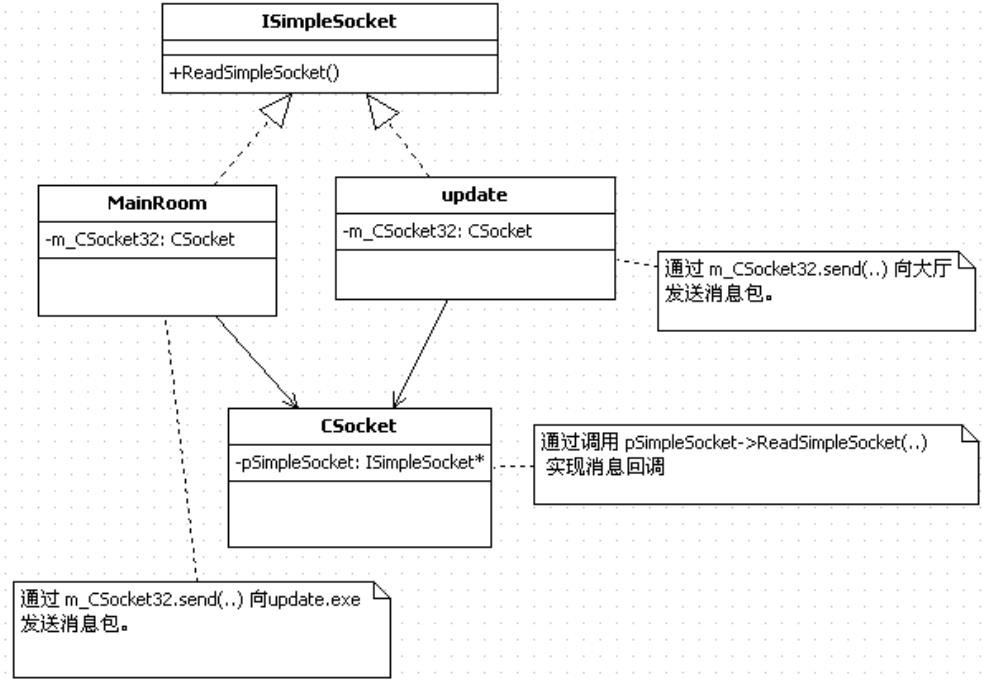
游戏大厅和更新升级程序的关系是服务器和客户端的关系，它们通过

socket连接。游戏大厅作为服务器端，更新升级程序作为客户端。它们是两个进程。当需要更新下载的时候，大厅启动更新程序(update.exe)，update.exe 启动后连接大厅服务器。这样，大厅就可以向 update.exe发送下载任务的消息包。Update.exe根据消息包的内容进行下载的动作。而且，大厅还可以继续向

update.exe发送任务消息包，从而实现多任务下载的功能。

封装一个 socket通信类，大厅和更新程序都分别建立一个 socket通信类实例对象来进行它们之间的通信。大厅和更新程序都继承一个 ISimpleSocket的接口，这个接口只有一个读消息的方法 ReadSimpleSocket(..)。这个方法就是它们读取对方消息的地方。

socket通信类里边维护了一个 ISimpleSocket\*指针。通过传入的大厅或更新程序指针初始化。这个指针通过调用 ReadSimpleSocket(..)，实现回调。

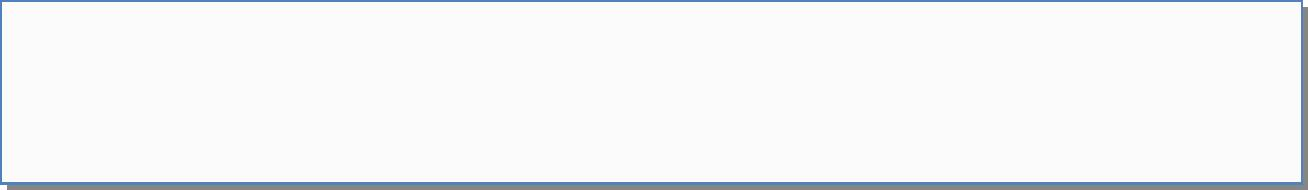


###### 大厅的更新

AServer.exe服务器会读取 HNGameGate.bcf 中更新服务器地址的配置。在大厅启动时读取这个地址，从而会形成一个下载 update\_version.txt的地址。大厅把包含此信息的消息包发给 update.exe，update.exe首先会下载这个文件。

其中，消息包包含下载地址，大厅版本，等必要信息。

update\_version.txt 的写法

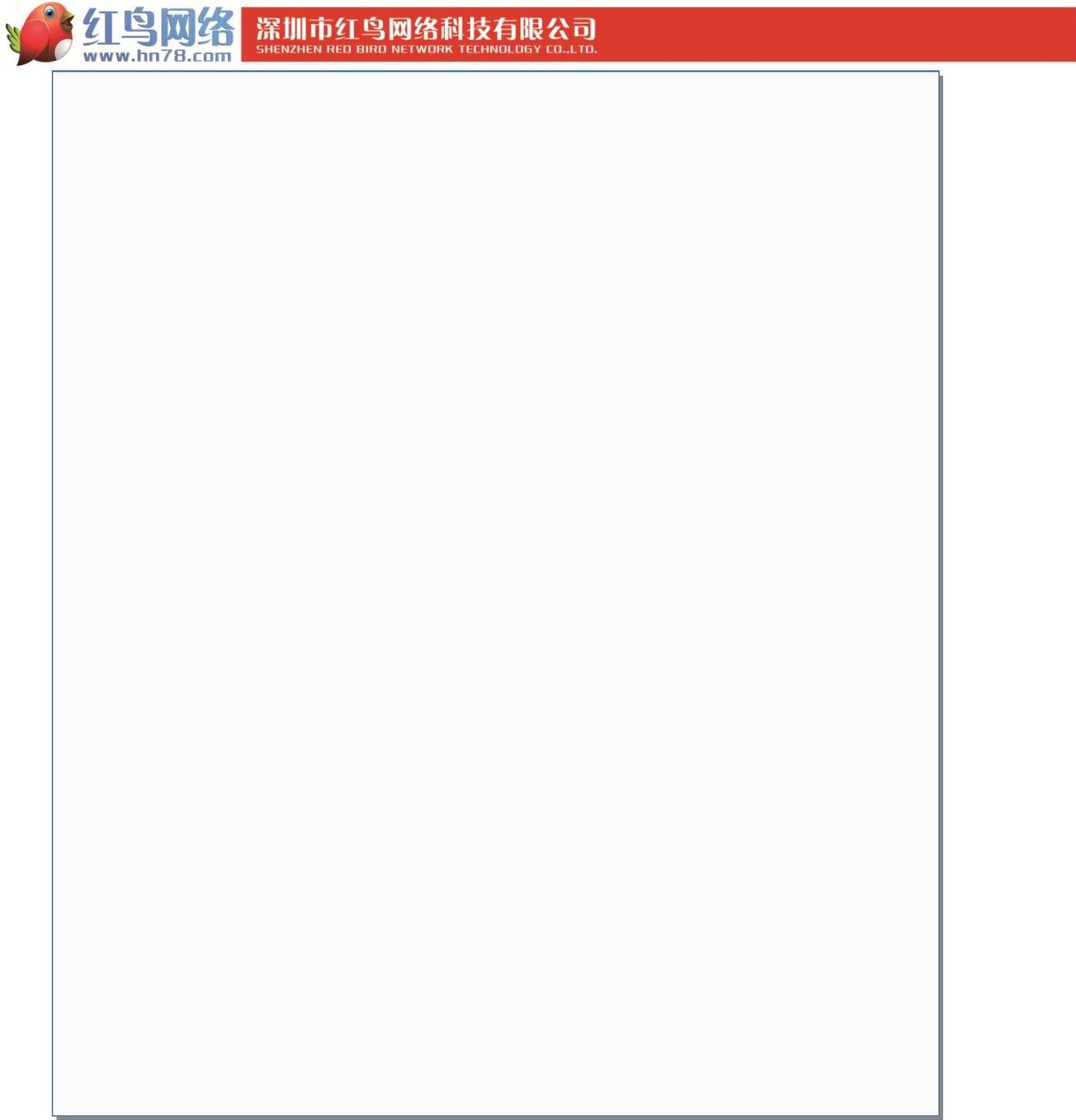


[config] hn.2.0=http://192.168.0.78:80/update/hn.2.0/downloadall.txt other=http://192.168.0.78:80/update/downloadall.txt

当 update\_version.txt被下载下来以后，update程序会先把消息中的版本号与 update\_version.txt的版本号比如:hn.2.0进行比较，如果有匹配的就下载那个版本的地址 http://192.168.0.78:80/update/

hn.2.0/downloadall.txt，如果没有就下载 other指向的地址。当然，可以写多个版本的地址连接，个数没有限制。这样就可以实现不同版本下载不同资源。

例如：

[config] hn.2.2=http://192.168.0.78:80/update/hn.2.2/downloadall.txt hn.2.3=http://192.168.0.78:80/update/hn.2.3/downloadall.txt other=http://192.168.0.78:80/update/downloadall.txt

2. downloadall.txt 写法

[list0] action=3

srcfile=http://192.168.0.78:80/update/游戏大厅.zip

destfile=游戏大厅.zip

cmd=DEC

[list1] action=3

srcfile=http://192.168.0.78:80/update/xxxxxx.txt destfile=xxxxxx.bat

cmd=EXEC

[list2] action=0 srcfile= destfile=

cmd=DEL/EXEC/REN/CRE

update 程序会分析 downloadall.txt，从[list0]开始，直到[listX]结束。

Action =0 表示结束

Action =1 表示 这个任务是要执行 cmd 对应的命令。

Action =2 表示 下载 srcfile 指向的文件。

Action =3 表示 下载+执行命令(1+2 的功能)

Destfile 指向的意思是下载下来以后的文件名。

cmd=DEC 表示解压 Destfile 文件

cmd=EXEC 表示运行 Destfile 文件

###### 游戏的更新

游戏的更新与大厅的更新本质是一样的。不同的是大厅更新在登录之前调用读取大厅的版本号，游戏更新是大厅登录之后，通过点击游戏列表读取游戏版本号。

例如：下载了 update\_version.txt后，如下内容。

解释：[list0]把 10100003.zip下载下来并解压，然后程序会把

10100003.zip删掉。[list1]下载一个 bat文件，并在本地运行它，完了以后 bat

把它自己删掉。[list2]表示完成

二.客户端程序设计思路

### 2. 1 大厅设计概述

大厅程序界面由各种组件与非组件窗口组装而成。大厅通过一个主窗口引组装子窗口，这些子窗口包括顶部窗口、左则窗口、及房间窗口等。有些子窗口又组装了更一步的子窗口，如左则窗口组装了游戏列表窗口、用户信息窗口。另外大厅有一些零散的窗口，如登陆框，公告弹窗等。

#### 2. 2 大厅进程类 CGamePlaceApp与大厅主窗口类 CGamePlaceDlg

1. CGamePlaceApp 为大厅进程的应用程序类。该类运行初始实例化函数

InitInstance()时便初始化大厅主窗口类 CGamePlaceDlg，启动大厅窗口。

1. CGamePlaceDlg::OnInitDialog 函数功能:
2. CGamePlaceDlg 初 始 产 生 CGameListWnd 类 窗 口 ( 游 戏 列 表 窗 口 ) 、CTopWndClass 类或 CTopDuWndClass 类窗口(顶部窗口)、CActWnd 类窗口(活动通知窗口)、CNewGameClass 类窗口(房间 TAB 控件窗口)。

.

4 CGamePlaceDlg::PreTranslateMessage 函数功能:

1. 处理消息用于该类窗口和 CTopDuWndClass 类窗口(顶部窗口)的最大化与还原。
2. 处理消息用于关闭该类窗口时关闭所有游戏房间
3. 处理消息用于重绘任务头像
4. 处理消息用于进出游戏房间时大厅背景音乐控制。

### 2. 3 大厅顶部窗口 CTopDuWndClass

1. 大厅顶部 CTopDuWndClass 窗口类为基于 DuiLib 开发的窗体。
2. CTopDuWndClass::OnCreate 函数功能: 创建及初始化窗体、控件
3. CTopDuWndClass::Notify 函数功能: 处理窗体控件的鼠标键盘消息等，处理

的消息及函数有:

* + 关闭按钮消息: 关闭游戏大厅。处理函数 OnHitClose
  + 最小化按钮消息: 最小化游戏大厅。处理函数 OnHitMin
  + 还原按钮消息: 还原游戏大厅。处理函数 OnHitRestore
  + 最大化按钮消息: 最大化游戏大厅。处理函数 OnHitMax
  + 换肤按钮消息: 弹出换肤窗体,调节窗体位置。处理函数 OnHitSkin
  + 其它窗体为编号从 1 开始的控件，可通过 top\TopWnd.xml 内的配置对应

的功能。有网页主页、银行、充值、商城、比赛、呼叫客服。处理函数

OnBnClickedBt1~11 OnBnClickedBt21~25

### 4 大厅游戏列表窗口 CGameListWnd

* 1. CGameListWnd::OnListMessage 函 数 功 能 :

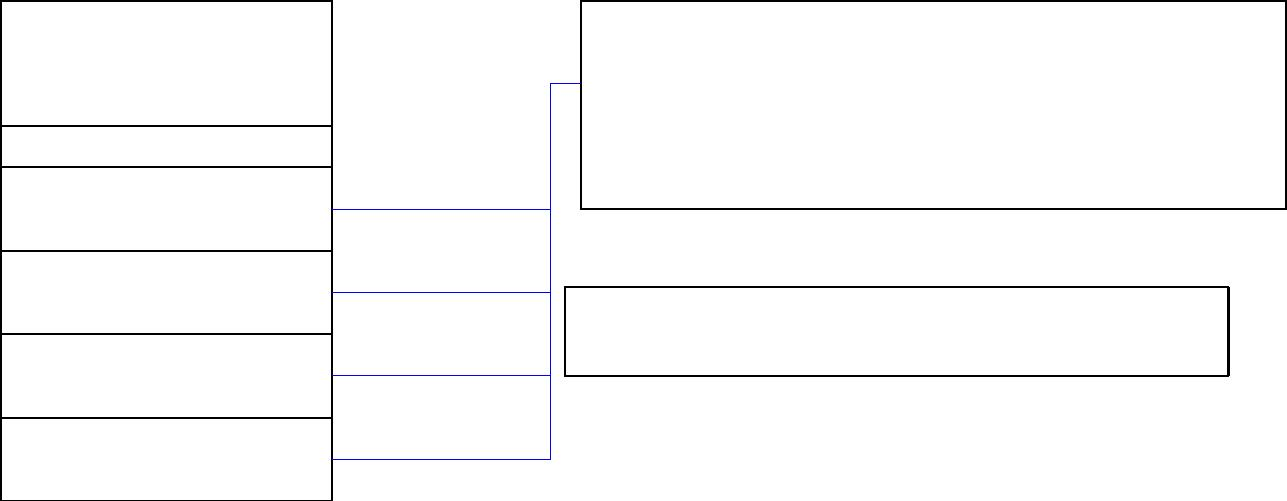
CMainRoomEx传窗体接收的消息。

* 1. 转发好友聊天的消息。
  2. 获取游戏类型并存入游戏链表。
  3. 获取游戏名字。
  4. 获取游戏人数列表。

### 5 大厅主程类 CMainRoomEx

处 理 主 进 程 窗 体 类

* + 1. 、CMainRoomEx与顶部窗口、游戏列表窗口的关系 (组合关系)；



CGamePlaceDlg

成 员 变 量 : CGameListWnd \*

CMainRoomEx(CGameListWnd\*,CActWnd\*, CTopWndClass\*,CTopDuWndClass\*)

CActWnd \*

CTopWndClass \*

CGamePlaceDlg的成员变量作为

CMainRoomEx构造函数的参数

CTopDuWndClass \*

* + 1. 、CMainRoomEx::EnterRoom函数实现进入游戏房间时的操作。 2.5.3、CMainRoomEx::PreTranslateMessage 处理鼠标键盘及自定义消息

等。处理的消息有:

* + 鼠标键盘消息
  + 根据用户 ID 获取用户昵称，或相反。消息 ID:

WM\_BANK\_GET\_NICKNAMEID

* + 银行操作相关的消息, 如存钱、转帐、修改密码、关闭银行等。
  + 购买道具请求. 消息 ID: WM\_USER+150
  + 呼叫客服相关请求.

CMainRoomEx::OnSocketReadEvent 函数处理网络读取的消息

* + MDM\_GP\_NETSIGNAL 网络信号消息
  + MDM\_CONNECT 连接消息类型
  + MDM\_GP\_REQURE\_GAME\_PARA 请求游戏全局参数
  + MDM\_GP\_LOGON 登录信息
  + MDM\_GP\_USERINFO 用户资料管理
  + MDM\_GP\_LIST 游戏列表消息转发
  + MDM\_GP\_IM 游戏好友聊天消息转发
  + MDM\_GP\_SYSMSG 系统消息
  + MDM\_GP\_MESSAGE 房间的消息 新闻
  + MDM\_GR\_USER\_LOGO 头像相关消息
  + MDM\_GP\_NOTICE\_MSG 提示消息
  + MDM\_GP\_PROP 用户道具消息
  + MDM\_GP\_LOCK\_ACCOUNT 请求锁定和解除锁定结果
  + MDM\_GP\_BIND\_MOBILE 绑定手机消息
  + MDM\_GP\_PLAYER\_2\_GM 普通玩家与 GM(管理员)间的通信
  + MDM\_GP\_RECEIVEMONEY 领取金币
  + MDM\_GP\_CHARMEXCHANGE 魅力值兑换
  + MDM\_GP\_MAIL 大厅邮件 公告 通知游戏等消息
  + MDM\_GP\_LOGONUSERS 统计登录人数(显示在线人数功能的消息代码)
  + MDM\_GP\_BANK 银行消息
  + MDM\_GP\_SMS 手机短信事务
  + MDM\_GP\_LASTLOINGGAME 向游戏列表添加游戏名称

### 6 房间设计概述

* 1. 从程序设计的角度，“房间”这个名词并不单纯指“游戏房间”，还包括

“主房间”。“游戏房间”顾名思义是看到游戏桌子、在线用户列表、聊天等

的那部分界面的集合；而“主房间”是用户未进入任何游戏房间时，所看到 的那个显示官方网站的界面。“主房间”与“游戏房间”的是通过一个 Tab 控件切换。如下图。



图:大厅界面各种窗口组装关系

* 1. 当用户登陆大厅后，不论进游戏房间与否，大厅程序首当其冲地建立一个 CMainRoomEx对象，而且在 CMainRoomEx的 Tab控件中插入一页，里面带有 WebBrowser 控件，显示官方网页。另外 CMainRoomEx 会初始化一个游戏房间指针类型的数组，但不创建游戏房间对象，数组的大小由 MAX\_GAME\_ROOM 宏决定。当用户从游戏列表进入第一个游戏房间，CMainRoomEx 会动态创建一个游戏房间的对象(CGameRoomEx)，并把其地址保存在游戏房间数组中。

Tab 页也新加一个页面，经过界面初始化后可以看到游戏桌子，在线用户列表等等游戏房间的内容。

* 1. CMainRoomEx 是红鸟大厅一个核心组件，他除了管理游戏的房间之处， 显示官方网页外还是大厅的部分网络消息在中转站。下面将详细房里的程序设计细节。

### 2. 7 房间的右则窗口 CRoomRightWnd

* + 1. CRoomRightWnd::OnBnClickedHideList 隐藏列表按钮
    2. 房间主程序 CGameRoomEx

CGameRoomEx:: PreTranslateMessage 处理鼠标键盘及自定义消息

* + - * WM\_BANK\_CLOSE 关闭银行
      * WM\_CHANGE\_PASSWD 银行发来的修改密码
      * WM\_TRANSFER\_RECORD 银行发来的转帐记录请求
      * WM\_TRANSFER\_MONEY 银行发来的转帐请求
      * WM\_CHECK\_MONEY 银行发来的存取钱消息
      * WM\_BANK\_GET\_NICKNAME\_ONID 银行界面发来的获取用户昵称的消息
      * WM\_TRANSFER\_ALL\_MONEY 银行界面发来的全转功能
      * WM\_BANK 银行消息
      * WM\_GAME\_LAUNCHED 游戏窗口已经启动完成
      * WM\_COLSE\_ROOM\_WITH\_MSG 关闭窗口消息
      * WM\_QUEUEROOM\_EXIT 退出排队机房间
      * WM\_CLOSEGAME 关闭游戏
      * WM\_GAME\_TIMEOUT 混战场时间到，关闭游戏
      * WM\_GM\_MSG G M 消息
      * WM\_GM\_KICK 踢出房间
      * 处理道具消息
      * 处理大小喇叭消息
    1. CGameRoomEx:: OnCreate 创建游戏房间
    2. CGameRoomEx::OnSystemMessage 处理系统消息
    3. CGameRoomEx::OnTimer 处理各个定时器的消息

### 2. 8 房间桌子 CDeskFrameView

* + 1. CDeskFrameView::InitDeskFramePng 初始化
    2. CDeskFrameView::SetViewDesk 设置视图桌子
    3. CDeskFrameView::CreateDeskFrame 建立游戏桌视图
    4. CDeskFrameView::UpdateDeskView 更新游戏桌
    5. CDeskFrameView::ShowBeginBtn 显示开始游戏按钮
    6. CDeskFrameView::ShowBJLBeginBtn 显示百家乐开始按钮
    7. CDeskFrameView::IsPlayGame 读取开始标志
    8. CDeskFrameView::GetUserItem 获取用户信息
    9. CDeskFrameView::SetUserItem 设置用户信息
    10. CDeskFrameView::GetNullCount 查找空位置数目
    11. CDeskFrameView::FindGameDesk 查找位置

CDeskFrameView::GetNullStation 获取空位置

* 1. 房间对游戏的控制类 **CLoveSendClassForEXE**

1. CLoveSendClassForEXE::StartGameClient 启动游戏 exe进程
2. CLoveSendClassForEXE::OnDestroy 停止游戏 exe进程
3. CLoveSendClassForEXE::AFCCreateFrame() 建立窗
4. CLoveSendClassForEXE::AFCCloseFrame() 关闭窗口
5. CLoveSendClassForEXE::SendGameData 向服务器发送消息
6. CLoveSendClassForEXE::SendWindowMessage 发送 WINDOWS消息
7. CLoveSendClassForEXE::OnControlMessage 控制消息，来自进程内部函数

的调用

1. CLoveSendClassForEXE::OnSocketReadEvent 网络读取消息
   * MDM\_GM\_GAME\_FRAME 框架消息 处理函数: OnFrameMessage
   * MDM\_GR\_PROP 道具消息 处理函数: OnPropMessage
   * MDM\_GR\_ROOM 房间消息

CLoveSendClassForEXE::OnFrameMessage 框架消息处理

CLoveSendClassForEXE::OnPropMessage 道具消息处理

* + ASS\_PROP\_USEPROP 道具消息 处理函数: OnUsePropResult

CLoveSendClassForEXE::OnChannelMessage 处理 IPC消息

* + IPC\_MAIN\_IPC\_KERNEL 内核消息 处理函数: OnIPCKernel
  + IPC\_MAIN\_SOCKET 网络消息 处理函数: OnIPCSocket
  + MDM\_GM\_GAME\_NOTIFY 游戏消息
  + MDM\_GM\_GAME\_FRAME 框架消息
  + IPC\_MAIN\_USER 用户消息 处理函数: OnIPCUserInfo
  + IPC\_MAIN\_CONFIG 配置消息
  + IPC\_MAIN\_PROP 道具主消息 处理函数: OnIPCUseProp
  + IPC\_PROP\_BUY\_VIP 道具界面购买 vip 处理函数: OnIPCBuyVIP
  + IPC\_FRIEND 加为好友
  + IPC\_EMENY 黑名单消息

1. CLoveSendClassForEXE::ShowMsgInGame 发送游戏房间消息
2. CLoveSendClassForEXE::SendIPCMessage 发送 IPC消息
3. CLoveSendClassForEXE::SendSocketMsg 发送网络消息
4. CLoveSendClassForEXE::SendGlbData 发送 Glb全局参数内容 13.CLoveSendClassForEXE::OnTimer 定时器消息处理函数

### 0 游戏的启动

CGameRoomEx::OnLeftHitDesk 函数响应点击座位操作，当满足进入游戏的条件和约束时，通过 socket向服务端发送消息(发送函数 Senddata, 辅助消息 ID:

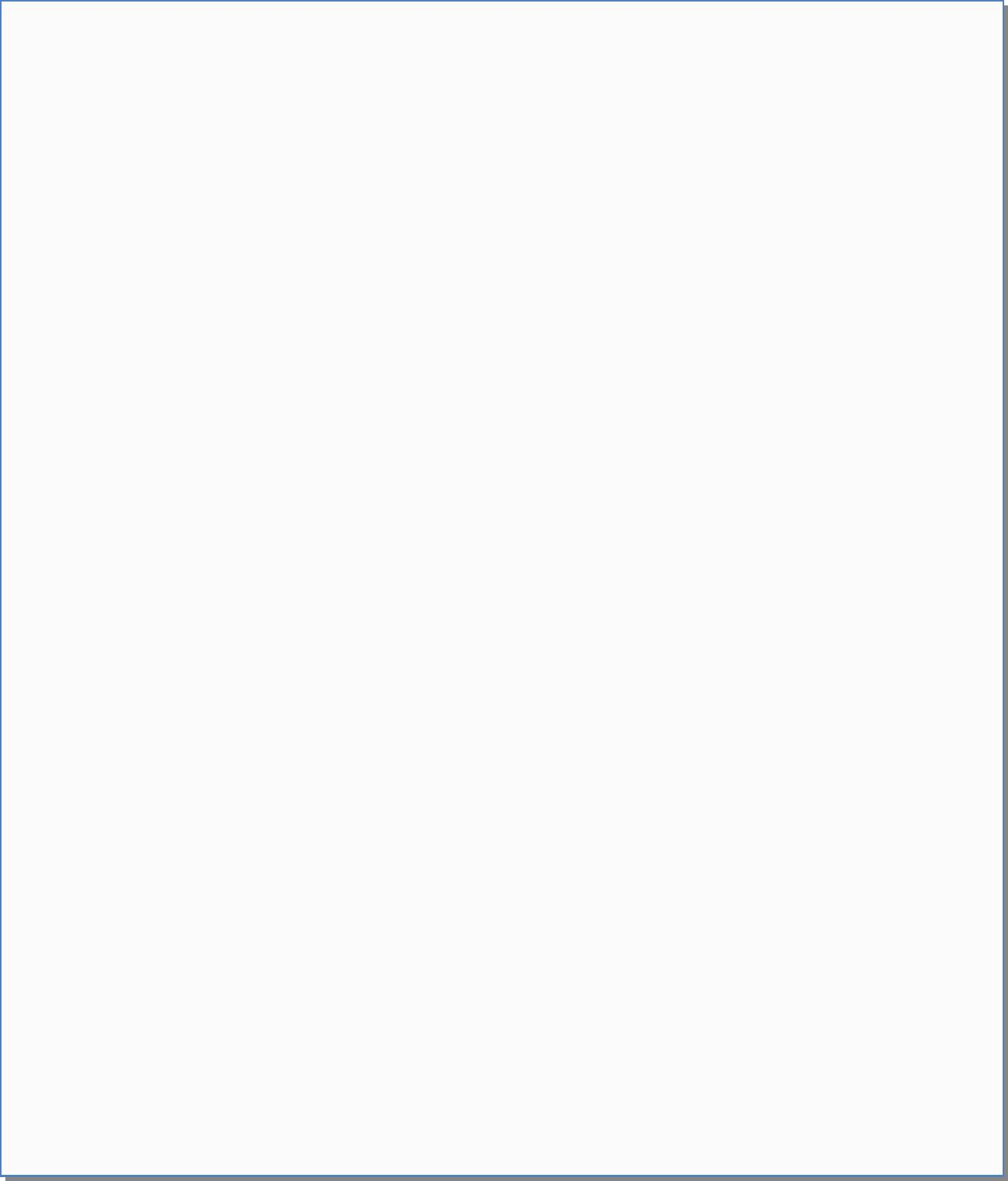
ASS\_GR\_USER\_SIT), 服务端接收到该消息后,更新该用户信息,然后给客户端返

回该消息,客户端 OnSocketReadEvent函数读取到该消息更新客户端显示并(通过 CLoveSendClassForEXE::::StartGameClient函数)启动游戏，启动游戏是通过 CreateProcess函数(输入参数有游戏 I D 等)创建 GameLauncher.exe进程。

关于 CreateProcess游戏进程的部分可参阅代码

CLoveSendClassForEXE::StartGameClient()，以下是代码段 StartGameClient

的启动游戏的主要部分。



……

/// 构造命令行

CString strCommonLine;

if (m\_pGameInfo->dwRoomRule & GRR\_GAME\_U3D)

{

m\_hWnd = NULL;

strCommonLine.Format(TEXT("%d\\%d.exe \"%d\""),m\_pGameInfo->uNameID, m\_pGameInfo->uNameID, m\_pSocket->GetSocketPort());

}

else

{

//从游戏信息里获得 EXE 程序的命令行

**strCommonLine.Format(TEXT("%d/%s /RoomToken:%s"),m\_pGameInfo->uNameID, "GameLauncher.exe",m\_szShareName);**

}

//启动游戏客户端STARTUPINFO StartInfo;

memset(&StartInfo,0,sizeof(StartInfo)); StartInfo.cb = sizeof(StartInfo);

StartInfo.wShowWindow = SW\_SHOWMAXIMIZED;

**BOOL bSuccess = CreateProcess(NULL,strCommonLine.GetBuffer(),NULL,NULL, FALSE,CREATE\_DEFAULT\_ERROR\_MODE,NULL,NULL,**

**&StartInfo,&m\_GameProcessInfo);**

strCommonLine.ReleaseBuffer(); if (!bSuccess)

{

memset(&m\_GameProcessInfo,0,sizeof(m\_GameProcessInfo)); return SM\_CREATE\_ERROR;

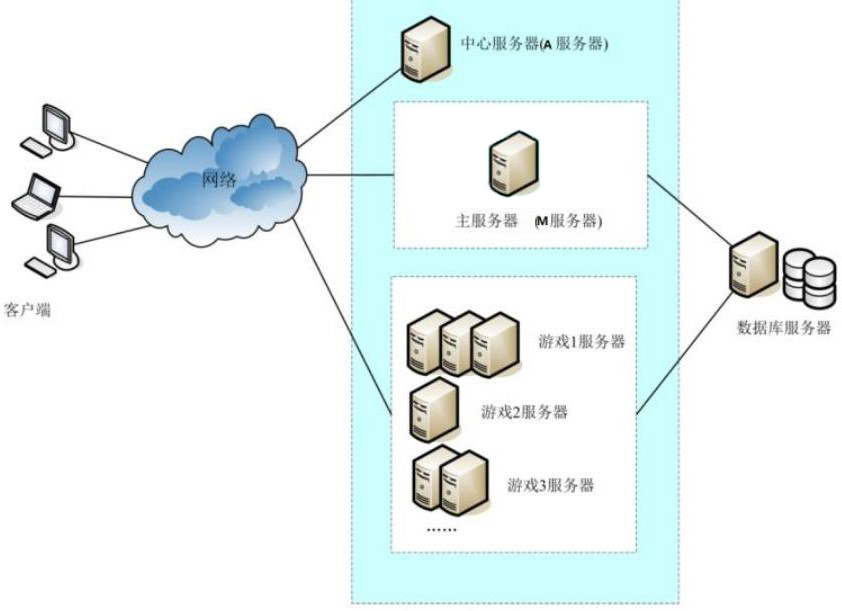
}

……

## 

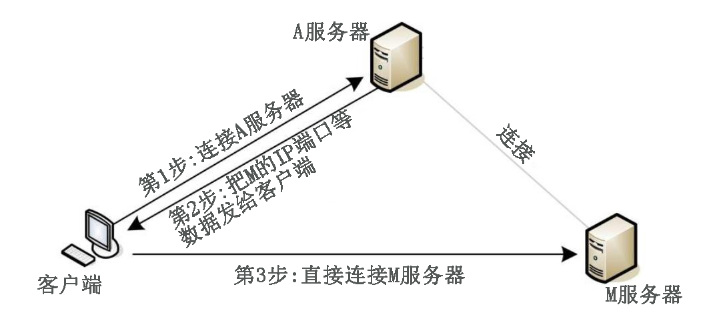
## 三.客户端与服务器的关系

### 1 红鸟服务器群落

* 1. 红鸟服务器支持分布服务，通过新建服务器可达到扩容的目的，形成红鸟服务器群落。无论服务器群落如何大，服务端的程序主要分为三类，中心服务器(A 服务器)、主服务器(M 服务器)、及游戏服务器(G 服务器)。三个服务器各自工作的侧重点。A 服务器提供接入；M 服务器平台基础服务(如登陆，游戏列表，好友系统等)；G 服务器负责游戏逻辑与房间坐位服务。
  2. 红鸟平台的正常运营，需要 A、M、G 服务器同时运行。而平台的基本运行，只需要 A、M 服务器即可。G 服务器属于增值扩充，在乎运营商购买了多少款游戏，没有 G 服务平台也运行，只是不能进行游戏。多个 G 服务器，但只有上一个 A 服务器和 M 服务器。仅有cocos客户端时，不需要开启A。M、G 服务器一方面对客户端的请求进行处理，别一方面需要与数据库交流。客户端的请求通过 M、G 服务器，M、G 进行逻辑处理，必要使用数据时M、G 从数据库中读写数据。目前 A 服务器不与数据库连接，A 服务器工作任务比较简单，也没有需要数据库的内容。
  3. 总体来说，A、M、G 服务器群落如下图所示

大厅界面各种窗口组装关系

### 2 客户端与 A 服务器（cocos客户端跳过此步骤）

* 1. 客户端启动首先接入 A 服务器，大厅的基础服务全在 M 与 G 服务器中， 客户端启动的最终目的是连接 M 服务器。AServer读取配置文件中设置的MServer地址及端口返回给客户端，客户端根据信息即可连接 M 服务器， 然后客户端可与 A 服务器断开接连。所以 A 服务器中客户端插入服务器的第一道门，其工作任务是十分简单的。工作步骤如下图所示：

### 3 客户端与 M 服务器

1. 在客户端连接 A 服务器时，客户端是什么登陆界面的，当收到 A 服务器发 M 服务器的信息，而断开 A 服务器时，登陆界面界面弹出。当用户输入用户名与密码后，这时客户端才连接 M 服务器。M 服务器连接成功并通过验证密码后，登陆界面才隐藏，大

厅才显示出来。

2. M 服务器为用户提供诸多基础服务，其中用户登陆登陆身份验证与游戏列表最为重要。用户登陆成功以后，M 服务器主动向该客户端发送一些基本数据， 如当前用户详细资料，游戏列表。以后的 M 服务大多靠客户端请求服务，如银行取钱，领取救济金等

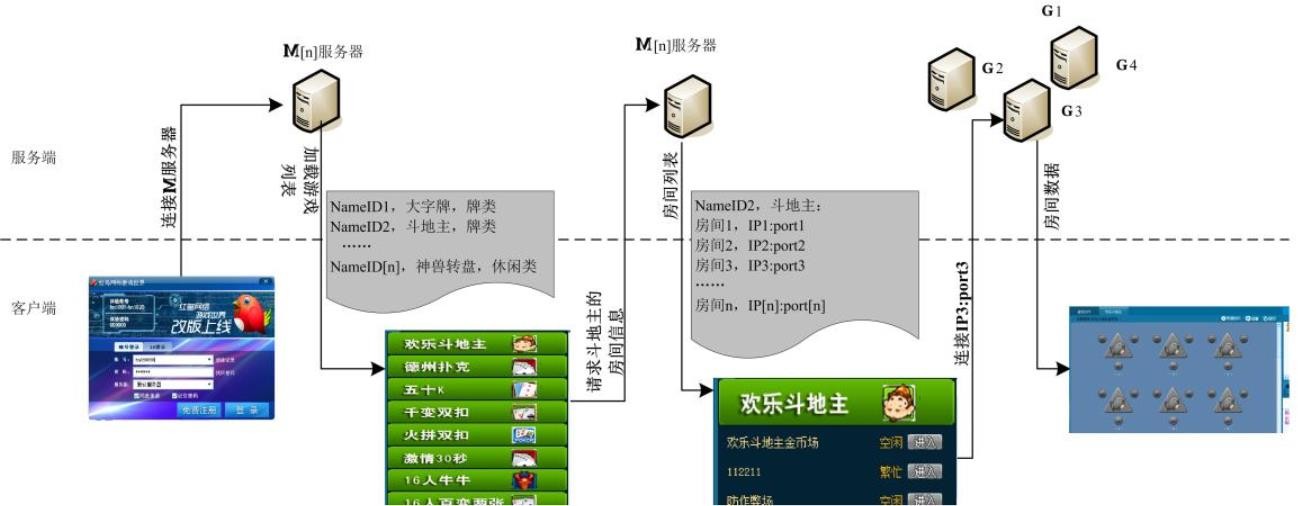
3. 客户端与 M 服务器的连接十分重要，如果 M 服务器关闭或网络中断， 客户端将强制关闭。

### 4 游戏房间与 G 服务器

* 1. 用户登陆大厅后，M 服务器后大厅发送游戏列表信息，这个游戏列表只包含游戏 ID、游戏名及游戏类型的信息。客户端利用这个数据组织起大厅左边的游戏列表界面。这个时间客户端程序只有游戏的信息并没有游戏对应的房间信息。那么游戏的房间列表是从何而来？答案是展开游戏列表，如下图。用展开击游戏列表的树形控件的游戏名时，客户端才向 M 服务器请对就游戏的房间列表。当客户端收到这个游戏房间列表数据，树形控件该游戏下就能显示出若干房间。
  2. 游戏房间列表数据包括房间名，房间 ID，及房间服务器(即游戏服务器或称 G 服务器)的 IP 与端口。这个数据包的房间的 IP 与端口尤其重要，关系到客户端如何连接 G 服务器，如何进入游戏房间。每个游戏房间的 IP 与端口数据绑定到房间列表中。当用户双击树形控件的游戏房间时，客户端取出该房间对应的 G

服务器的 IP 与端口，创建一个新的 socket 连接之。同样，如果平台支持可同时

进多个房间。那么每进一个游戏房间就多建一个 socket 连接对应的 G 服务器。所以整个平台的与服务器普通运行时的网络连接数为 1 个恒连接 M 服务器的

socket 加上若干个连接 G 服务器的 socket。

* 1. 连接 G 服务器的 socke，随退出房间而关闭。
  2. 如果一个游戏曾经加载过房间列表，则在用户退出前，再次点击游戏树形列表中的游戏名字，则不予以向 M 请求。那么说，游戏房间列表再每次登陆后只获取一次。所以如果 G 的房间有增减及修改，A 与所有的 M 服务器都需要重启，客户端也需要重启，否则客户端获取的房间列表仍然是增减修改前的数据。

### 5 红鸟的网络消息包的结构

###### 大厅消息

* MDM\_GP\_NETSIGNAL 网络消息
* MDM\_GP\_BANK (主 ID) 银行消息 ASS\_GP\_BANK\_TRANS\_RECORD (辅助 ID)

转帐记录

* MDM\_GP\_LOGON (主 ID) 登录信息 ASS\_GP\_LOGON\_LOCK\_VALID (辅助 ID)

机器被锁定

* MDM\_GP\_USERINFO (主 ID) 用户资料管理 ASS\_GP\_USERINFO\_ACCEPT 接受用户消息
* MDM\_GP\_SYSMSG (主 ID) 系统消息 ASS\_GP\_NEWS\_SYSMSG 新闻
* MDM\_GP\_PROP 用户道具消息 ASS\_PROP\_BUY 购买道具

对应银行，登录，用户信息，系统消息，道具功能模块。

###### 房间消息

大至按用户动作消息，游戏消息，框架消息等进行分类网络消息

* MDM\_GP\_NETSIGNAL 网络信号消息
* MDM\_GR\_MONEY 钱柜划账 扣钱 ASS\_GR\_TRANSFER\_RECORD 转帐记录
* MDM\_GR\_LOGON 登录 ASS\_GR\_LOGON\_SUCCESS 登录成功
* MDM\_GR\_USER\_LIST 用户列表 ASS\_GR\_ONLINE\_USER 在线用户
* MDM\_GR\_USER\_ACTION 用户动作消息 ASS\_GR\_USER\_UP 用户起来

###### 大厅主程 CMainRoomEx的网络消息收发

发送消息:

int CTCPClientSocket::SendData(void \* pData, UINT uSize, BYTE bMainID, BYTE bAssistantID, BYTE bHandleCode)

接收消息:

boolCMainRoomEx::OnSocketReadEvent(NetMessageHead\*pNetHead,void

\* pNetData, UINT uDataSize, CTCPClientSocket \* pClientSocket)

###### 房间主程 CGameRoomEx的网络消息收发

发送消息:

int CTCPClientSocket::SendData(void \* pData, UINT uSize, BYTE bMainID, BYTE bAssistantID, BYTE bHandleCode)

接收消息:

boolCGameRoomEx::OnSocketReadEvent(NetMessageHead\*pNetHead,void

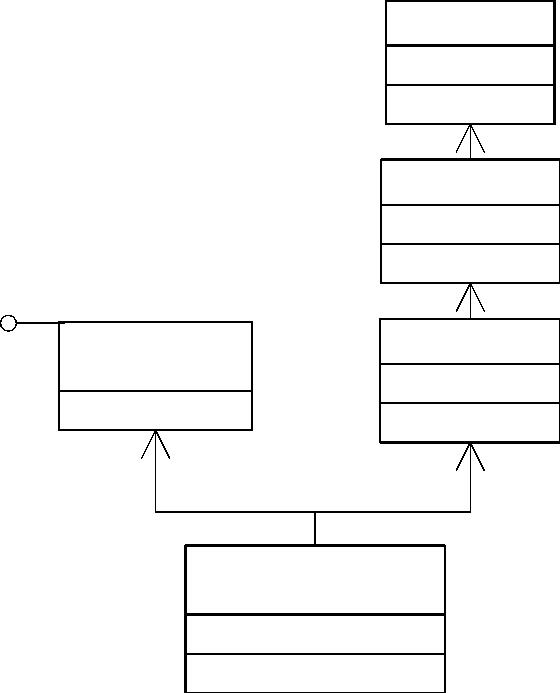
\* pNetData, UINT uDataSize, CTCPClientSocket \* pClientSocket)

### 四.房间与游戏进程的关系

#### IFrameInterface接口

* + 1. GameLauncher 工程中提供了IFrameInterface 接口,而CLoveSendClassInExe

类继承自 IFrameInterface 通过虚函数实现接口功能。



**CDialog**

**CGameFaceGo**

**CLoadFileBmp**

<<接口>>

**IFrameInterface**

<<实现类>>

**CLoveSendClassInExe**

提供给游戏的接口

* + 1. OnControlMessage 控制消息
    2. OnSocketReadEvent 网络消息
    3. ShowMsgInGame 发送游戏房间消息
    4. ShowMsgDudu 发送嘟嘟消息
    5. CLoveSendClassInExe 类对应的实现:
    6. CLoveSendClassInExe::OnControlMessage
* IPC\_USER\_PROP\_RESULT 道具操作结果消息
* CM\_USER\_SEND\_TIMES\_MONEY 用户获取按次赠送的金币
* CM\_USER\_STATE 用户状态改变
* IPC\_TROTTING 跑马灯消息
* IPC\_EX\_SKIN 换肤消息

#### 大厅的房间与游戏两进程的底层通讯方式

大厅与房间两进程间的底层通讯方式是建立文件映射内核对象，建立文件共享内存。从而实现进程间的通讯。

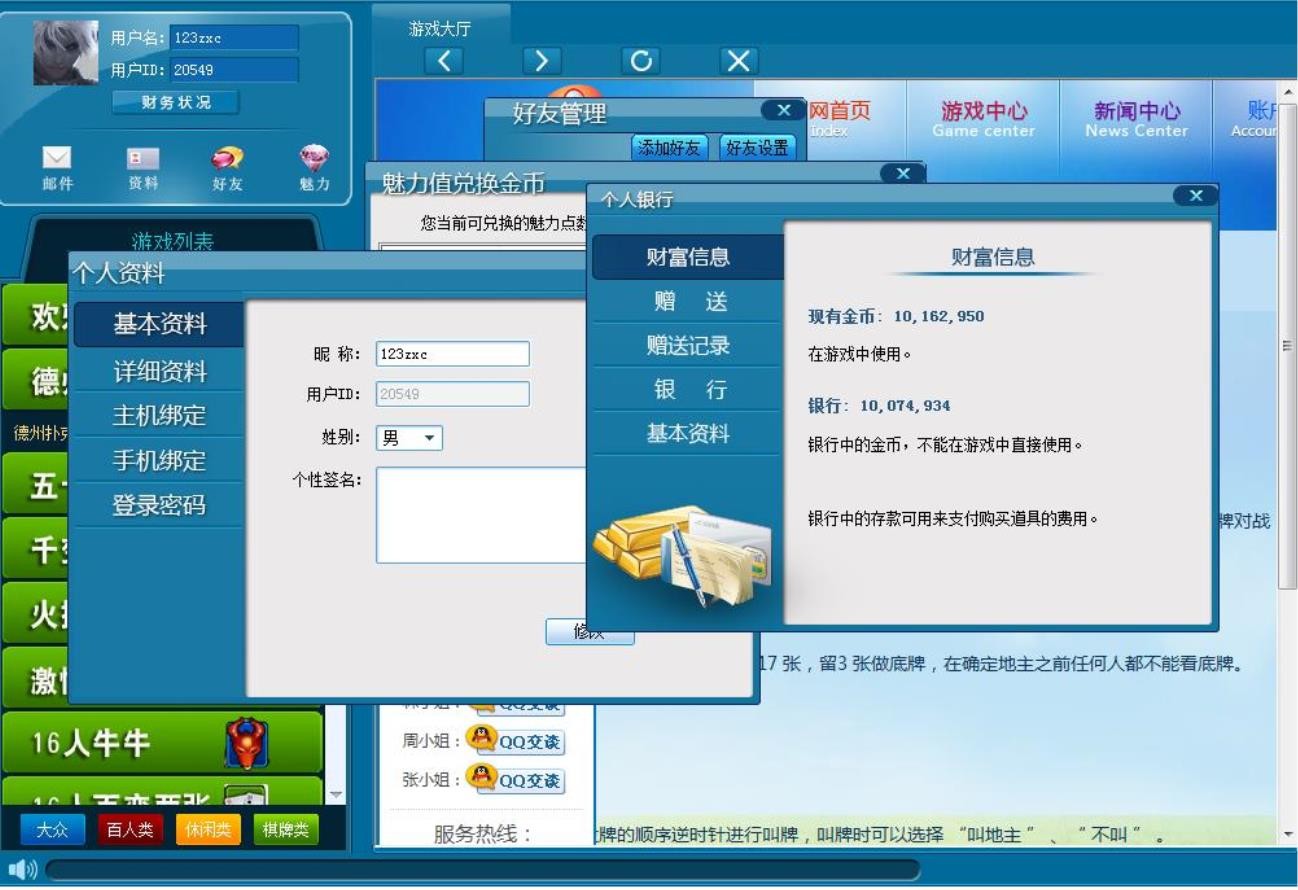
#### 3 房间与游戏几个重要的 I P C 消息

* 网络消息 IPC\_MAIN\_SOCKET
* 配置信息 IPC\_MAIN\_CONFIG
* 用户信息 IPC\_MAIN\_USER
* 控制信息 IPC\_MAIN\_CONCTROL
* 道具主消息 IPC\_MAIN\_PROP

### 五.红鸟产品的界面技术

###### 1 红鸟主流界面开发技术概述

红鸟主流界面开发技术有： DirectUI、及 Unity3D、HGE引擎。

* 1. DirectUI是新引进的 U I 技术，使用开源 U I 库 DUIlib改造而来，DirectUI 意为直接在父窗口上绘图(Paint on parent dc directly)。BzDirectUI 拥有 U I 编辑器。可以很方便的构建高效，绚丽的，非常易于扩展的界面，易于实现各种超炫的界面效果如换色，换肤，透明等。与百宝湾 U I 异曲



同工。目前多用于平台的界面，逐渐取代 MFC的界面开发技术。

* 1. Unity3D引擎游优点是优越的游戏体验好、可以实现 3 D 游戏效果，缺点

是开发周长，文件体积大。如 3 D 森林舞会。

* 1. HGE 引擎是一个硬件加速（Hardware accelerated）的 2D 游戏引擎（Game



Engine），HGE 是一个富有特性的中间件，可以用于开发任何类型的 2D 游



戏。HGE 封装性良好，以至于你仅仅需要关心游戏逻辑（Game Logic），而



不需要在意 DirectX，Windows 消息循环等。HGE 架构在 DirectX 8.0 之



上，能够跑在大多数的 Windows 系统上。

### 六.平台服务端结构组成概述

###### 服务端主要组成模块及其功能

* + 1. AServer.exe:中心服务器， M 服务器管理，大厅版本升级
    2. MServer.exe：登录服务器，登录，游戏列表，好友，银行，道具，邮件

等大厅功能

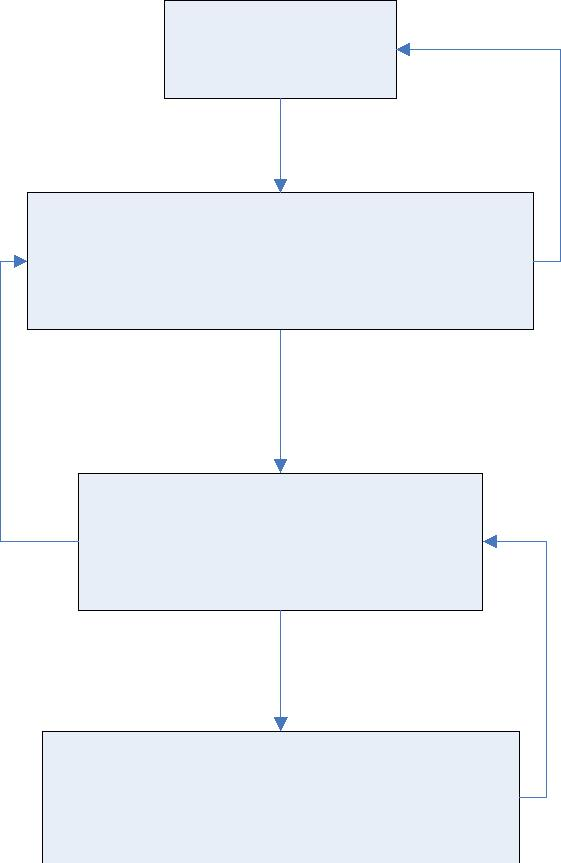
* + 1. GServer.exe：房间管理服务器，加载房间组件；
    2. AFCComDllEx.dll：公用库，数据库，网络等公用模块；
    3. GameManage.dll：房间组件，游戏服务端组件管理，服务器主要负责房间登录，桌子列表；
    4. 用户列表，银行，道具（使用），房间内的聊天，用户房间内动作包括（坐

下，起来，同意等），游戏流程（开始，结束，结算等），排队机房间的除外。比赛场的除外等.

* + 1. CfgEngine.dll：\*.bcf文件操作库
    2. HNlibSqlEngine.dll：数据库操作封装库
    3. AFCComEx模块的功能
    4. MServer模块的功能
    5. GameManage模块的功能
    6. GServer模块的功能
    7. AServer模块的功能

### 七.平台服务端程序设计思路

###### 服务端基本框架



客户端数据

网络模块（可视为单独线IOCP)

基类：CTCPSocketManage

逻辑模块

基类：CBaseMainManage

数据库模块

基类：CAFCDataBaseManage

如图所示：

* 1. 网络模块：**CTCPSocketManage**
     1. 负责接收来至客户端的消息，使用完成端口实现（实际上就是利用了系统实现的可以处理异步 IO 的数据队列） 详细的过程就是系统会把收到的 TCP 消息，放到指定的地方。读网络数据的线程在有完整的 TCP包时会收到系统的通知， 去指定的地方取数据合并给完整的数据包后， 再调用逻辑模块的

OnSocketRead将数据包传递给逻辑模块来处理。

* + 1. 网络模块最主要的类为：CTCPSocket，CTCPSocketManage等
  1. 逻辑模块基类：**CBaseMainManage**
     1. 有 客 户 端 消 息 需 要 处 理 时 ， 网 络 模 块 首 先 调 用 逻 辑 模 块 的OnSocketRead，进行相应的处理（包括判断包的合法性，还有一系列的逻辑处理），如果不需要进行数据库进行操作，就将处理结果直接通过网络模块发送给客户端。否则就将数据库操作指令添加到数据库操作队列中去。当数据库模块的读取线程调用到相应指令后，就会根据指令的内容去操作存储过程，当存储过程的值返回后调用逻辑模块 OnDataBaseResult，逻辑模块再将存储过程返回的结果进行进一步的处理，再将最终结果通过网络模块发送给客户端。
     2. 逻辑模块最主要的类为：CBaseMainManage，CBaseMainManageZ等

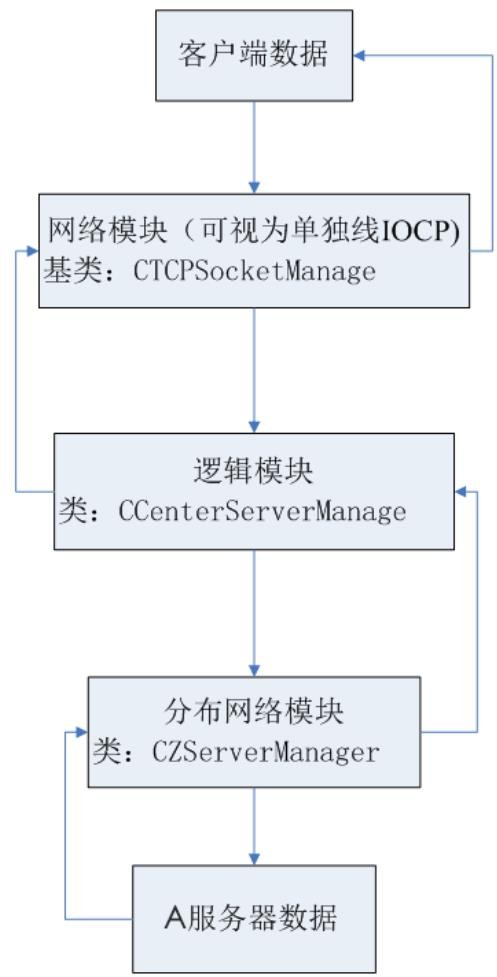
###### 4 数据库模块基类：CAFCDataBaseManage

* 1. 单独一个线程，PushLine接口提供给逻辑模块添加指令加队列中，当队列中有数据时，根据指令内容执行相应的存储过程，并获取存储过程执行完毕后返回的结果。然后调用逻辑模块 OnDataBaseResult函数将结果传递给逻辑模块来处理。
  2. 数据库模块最主要的类为：CAFCDataBaseManage，CDataBaseHandle，

CDataLine等

* 1. 每个服务器的实现基本分为两步：
  2. 首先是要继承 CBaseMainManage 或者 CBaseMainManageForZ，最主要的目的就是实现其中的 OnSocketRead函数，以及 OnDataBaseResult函数
  3. 继承 CAFCDataBaseManage，主要是实现 HandleDataBase函数

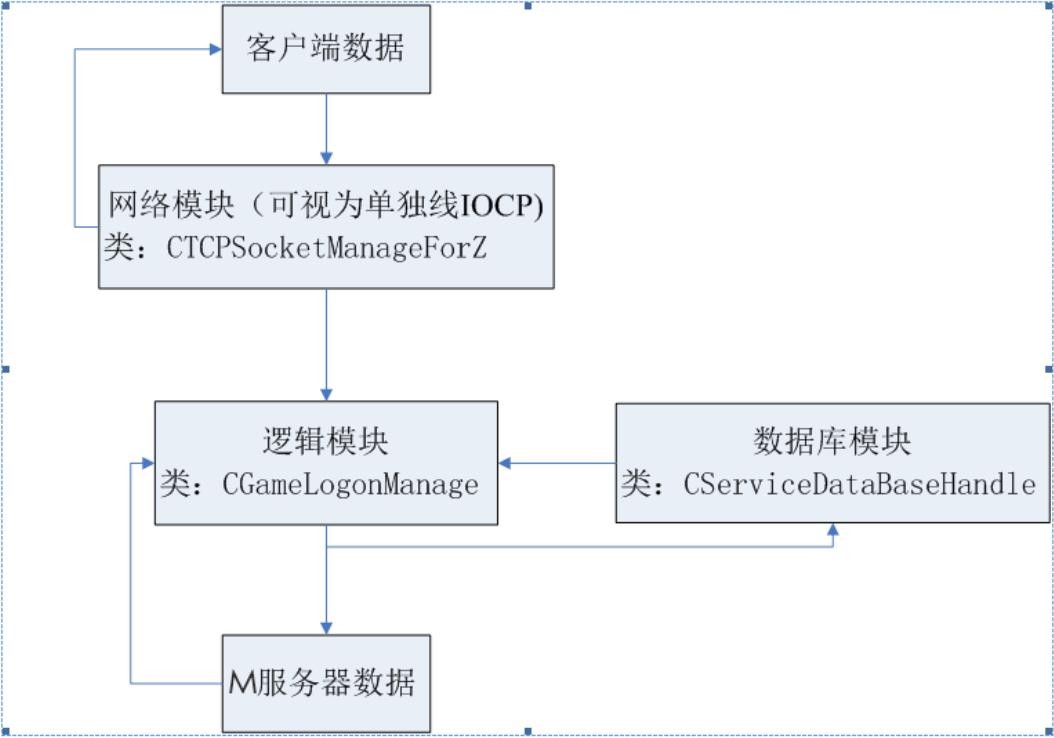
###### Aserver框架



* + 1. A 服务器的实现为如上图所示：
    2. CCenterServerManage类继承 CBaseMainManage
    3. 因为 A 服务器并不需要处理数据库所以不用继承 CAFCDataBaseManage

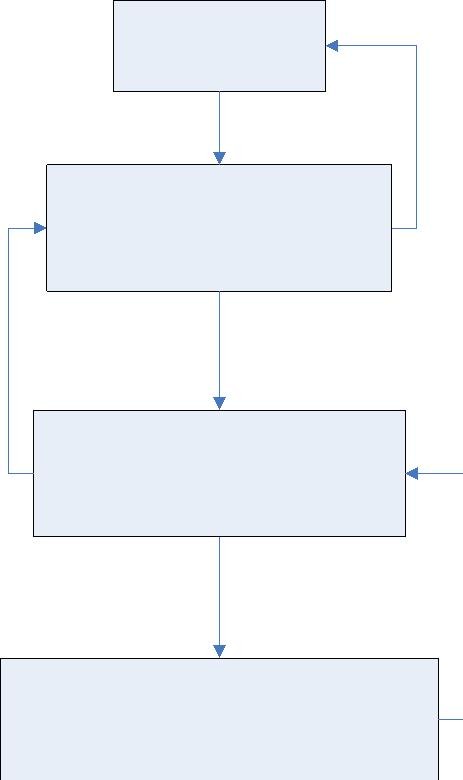
来重写 HandleDataBase函数

* 1. **MServer** 框架



* + 1. M 服务器的实现为如上图所示：
    2. CGameLogonManage类继承 CBaseMainManage
    3. CServiceDataBaseHandle类继承 CAFCDataBaseManage

##### GameManage 框架



客户端数据

网络模块

类：CTCPSocketManage

逻辑模块

类：CGameMainManage

数据库模块

类：CGameDataBaseHandle

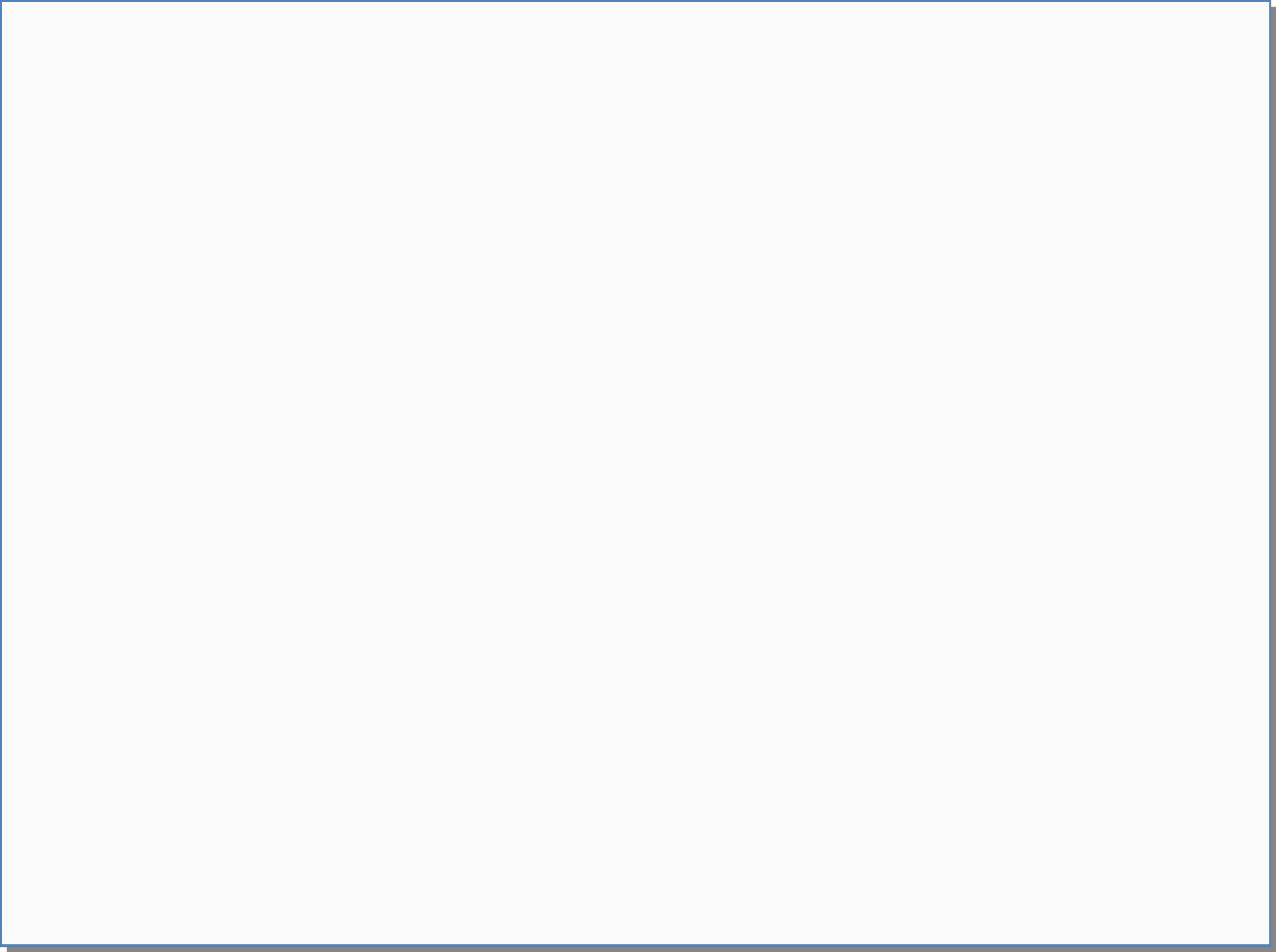
* + 1. G 服务器的实现为如上图所示：
    2. CGameMainManage类继承 CBaseMainManage
    3. CGameDataBaseHandle类继承 CAFCDataBaseManage

###### 7. 8 服务端的基本数据流程

服务端重要的类及其关系：

### 八. GServer与游戏服务端的调用

* 1. **CGameDesk** 提供给游戏服务端的接口



//是否可以开始游戏

virtual bool CanBeginGame();

//是否可以断线

virtual bool CanNetCut(BYTE bDeskStation);

//用户坐到游戏桌

virtual BYTE UserSitDesk(MSG\_GR\_S\_UserSit \* pUserSit, CGameUserInfo \* pUserInfo);

//用户离开游戏桌

virtual BYTE UserLeftDesk(BYTE bDeskStation, CGameUserInfo \* pUserInfo);

//旁观用户坐到游戏桌

virtual BYTE WatchUserSitDesk(MSG\_GR\_S\_UserSit \* pUserSit, CGameUserInfo \* pUserInfo);

//旁观用户离开游戏桌

virtual BYTE WatchUserLeftDesk(BYTE bDeskStation, CGameUserInfo \* pUserInfo);

//用户同意开始

virtual bool UserAgreeGame(BYTE bDeskStation);

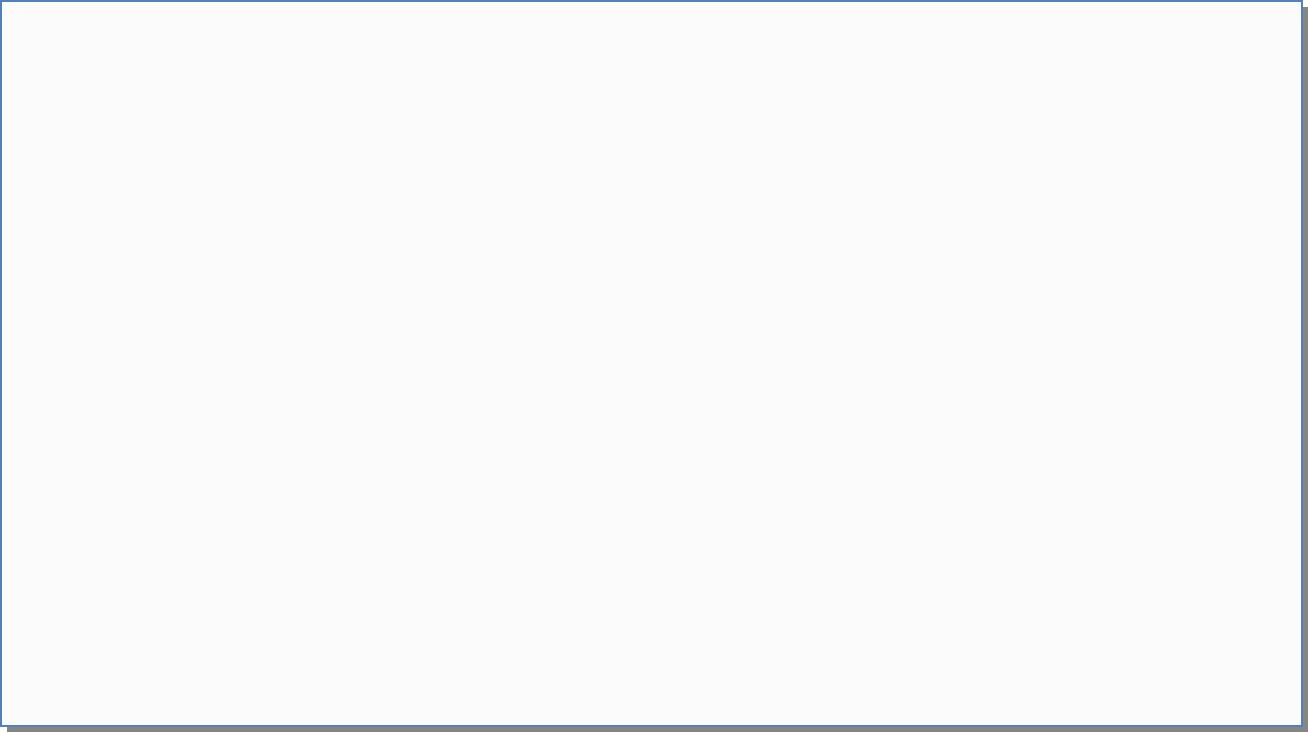
//框架消息处理函数

virtual bool HandleFrameMessage(BYTE bDeskStation, NetMessageHead \* pNetHead, void \* pData, UINT uSize, UINT uSocketID, bool bWatchUser);

//旁观参数变化

virtual bool OnWatchSetChange(BYTE bDeskStation, long int dwTargetUserID, bool bEnableWatch)

###### GServer装载游戏服务端 dll的方法



pComRoomInfo->hDllInstance=AfxLoadLibrary(pComRoomInfo->ServiceInfo.szDLLFileName); if (pComRoomInfo->hDllInstance==NULL)

return false;//throw TEXT("组件装载失败");

//获取管理接口

CreateServiceInterface \* pCreateServiceInterface=(CreateServiceInterface \*) GetProcAddress(pComRoomInfo->hDllInstance,TEXT("CreateServiceInterface"));

if (pCreateServiceInterface==NULL)

return false;//throw TEXT("非法组件");

pComRoomInfo->pIManageService=pCreateServiceInterface(DEV\_LIB\_VER); if (pComRoomInfo->pIManageService==NULL)

return false;//throw TEXT("组件管理接口获取失败，可能是资源不足");

GetServiceInfo \* pGetServiceInfo=(GetServiceInfo\*)

GetProcAddress(pComRoomInfo->hDllInstance,TEXT("GetServiceInfo"));

九.服务端相关的数据库

* 1. AServer服务端使用的主要表
     + TUserProp
     + TPropDefine
     + TUsers
     + TUserInfo
     + TZLoginRecord
     + TGameRoomInfo

###### AServer服务端使用的主要存储过程

* + - SP\_UserLoginPlace
    - SP\_GetOnlineRoom

###### GServer服务端使用的主要表

* + - TGameRoomBase
    - TGameRoomInfo

###### GServer服务端使用的主要存储过程

* + - SP\_GetRoomList

###### 9

1. GameManage服务端使用的主要表
   * TGameRoomBase
   * TGameRoomInfo
   * TContestInfo
   * TContestUserRecord
   * TUsers
   * TUserInfo
   * TWLoginRecord

###### GameManage服务端使用的主要存储过程

* + - SP\_UserLoginRoom
    - SP\_UserInfoUpdateAll