

**天天爱消除**

**PC 版**

# **软件架构设计文档**

队长：121250151 王 琨

队员：121250121 沈静怡

121250127 石文磊

121250101 缪晓伟

121250156 王天宇

121250167 吴晓晨

# 修订历史记录

修改人员	日期	版本描述	版本号
缪晓伟	2014. 4. 17	最初草稿，发布 至 SVN 供参考	V1. 0
缪晓伟	2014. 4. 20	修正细节	V1. 1

# 目录

1.引言.....	4
1.1 编写目的.....	4
1.2 对象与范围.....	4
1.3 参考文献.....	4
2.体系结构模型.....	4
2.1 整体架构描述.....	4
2.2 运行时进程.....	6
2.3 物理部署.....	7
2.4 架构设计.....	8

# 1.引言

## 1.1 编写目的

本文档提供天天爱消除 PC 版的软件架构概览,采用若干架构视图描述系统的不同方面,以便表示构造系统所需要的重要架构决策。

## 1.2 对象与范围

本文档的读者是 Ex 咖喱棒团队内部的开发和管理人员,参考了 IEEE 的系统设计文档模版和 RUP 的软件架构文档模版,用于指导下一循环的代码开发和测试工作。

## 1.3 参考文献

- (1) IEEE 标准
- (2) 天天爱消除 PC 版项目范围文档
- (3) 天天爱消除 PC 版需求规格说明文档
- (4) 天天爱消除 PC 版系统测试用例文档

# 2.体系结构模型

## 2.1 整体架构描述

天天爱消除 PC 版选择分层的体系结构,将系统分为展示层、逻辑层和数据层。展示层包含 GUI 的实现和人机交互,逻辑层用于控制各个模式游戏的逻辑判定和系统内部简单逻辑,数据层用于保存持久化数据(玩家的游戏历史数据)和数据的基本读写逻辑。

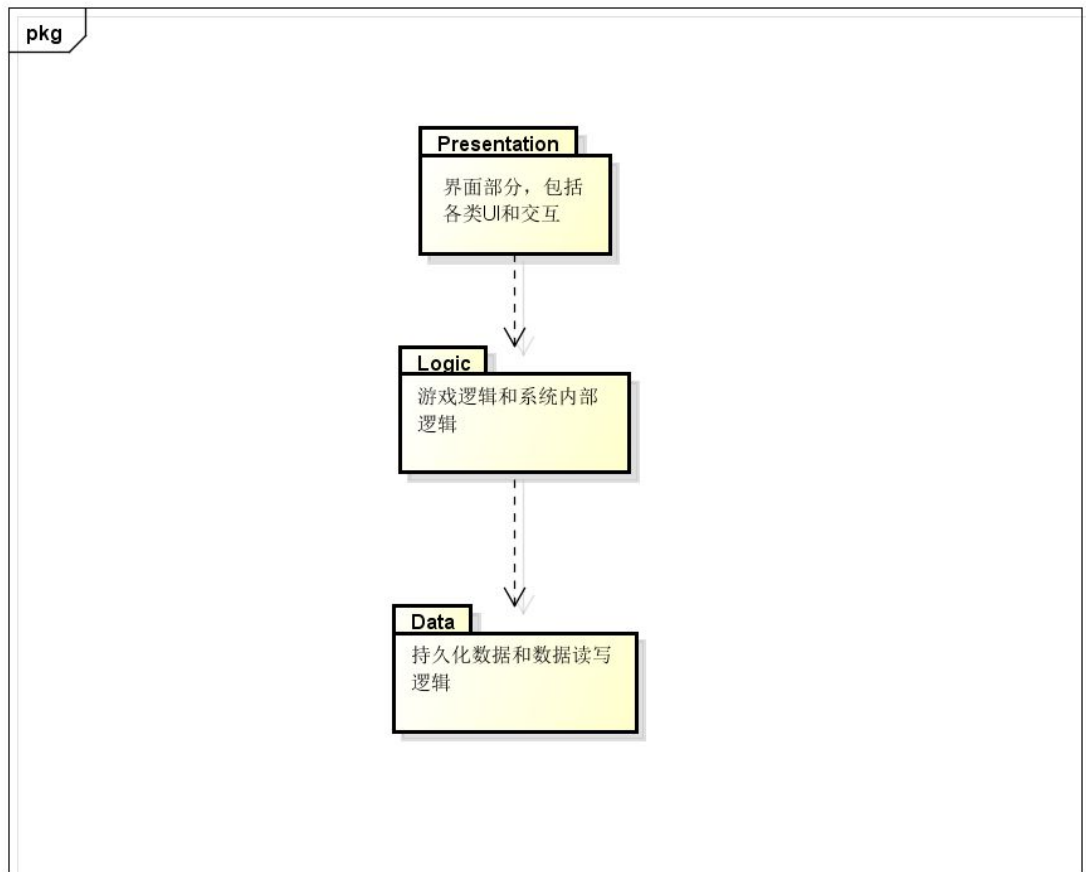


图 1 客户端的体系结构逻辑表达

在客户端以外需要增设服务端，用于对战/协作模式下不同客户端间的游戏信息转发和玩家统计数据的保存，以生成排行榜。

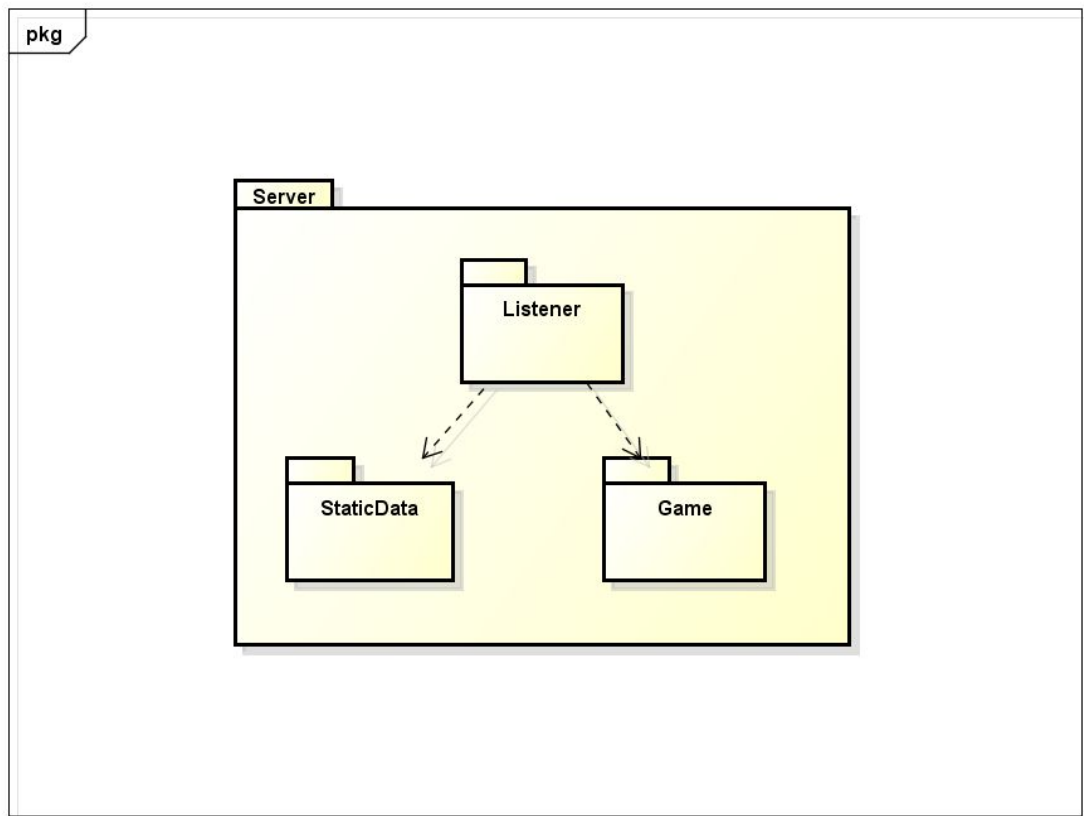


图 2 服务端的体系结构逻辑表达

## 2.2 运行时进程

在同一个局域网中，会有天天爱消除 PC 版的一个或多个客户端进程和一个服务端进程，其进程图如图所示。客户端进程在客户端机器上运行，服务器进程可与客户端进程在同一台机器上运行，也可在单独的服务端机器上运行。

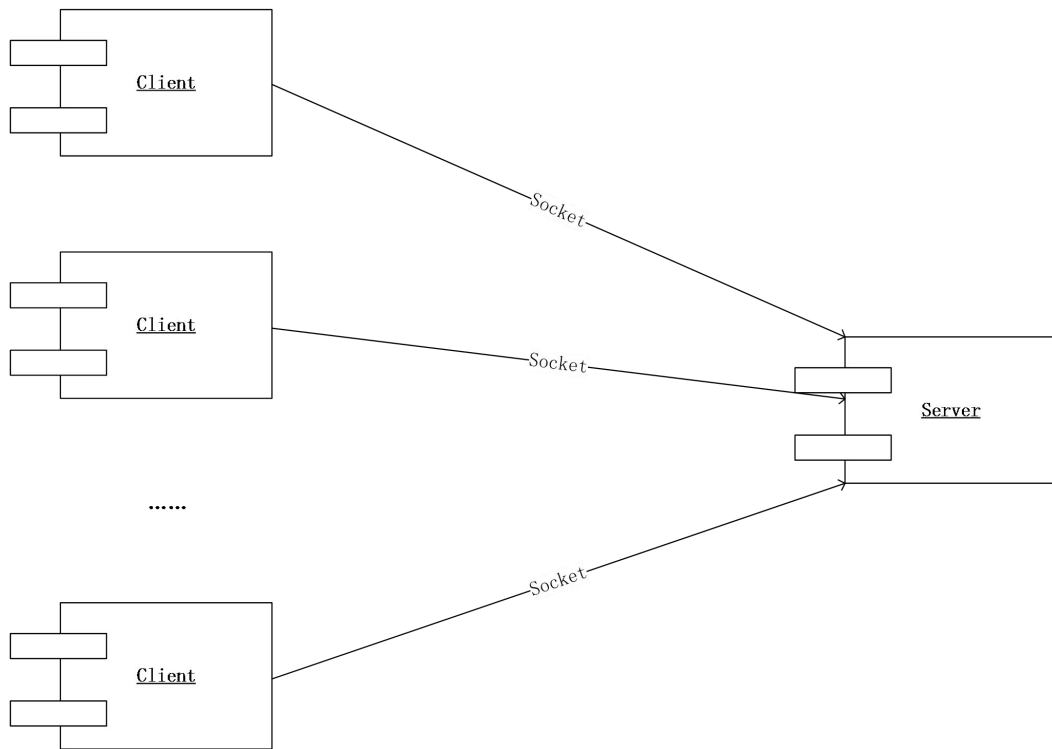


图 3 进程图

## 2.3 物理部署

天天爱消除 PC 版的客户端放在客户端机器上，服务端放在服务端机器上。由于客户端与服务端之间的通信采用 Java 网络组件中原生支持的 Socket，因此在客户端和服务端的 JDK/JRE 环境正确配置的情况下不需要另外进行部署。

系统可以部署在以下 2 个物理层次。

- (1) 访问层：用于用户使用系统的层次；
- (2) 服务层：部署服务器和服务器相关统计数据的层次。

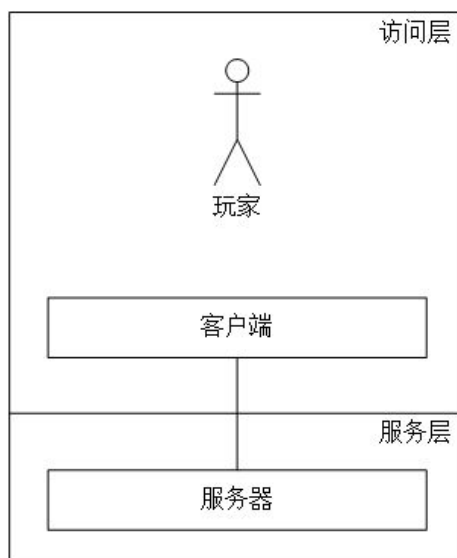


图 4 系统的部署层次

## 2.4 架构设计

服务端的架构设计如下。

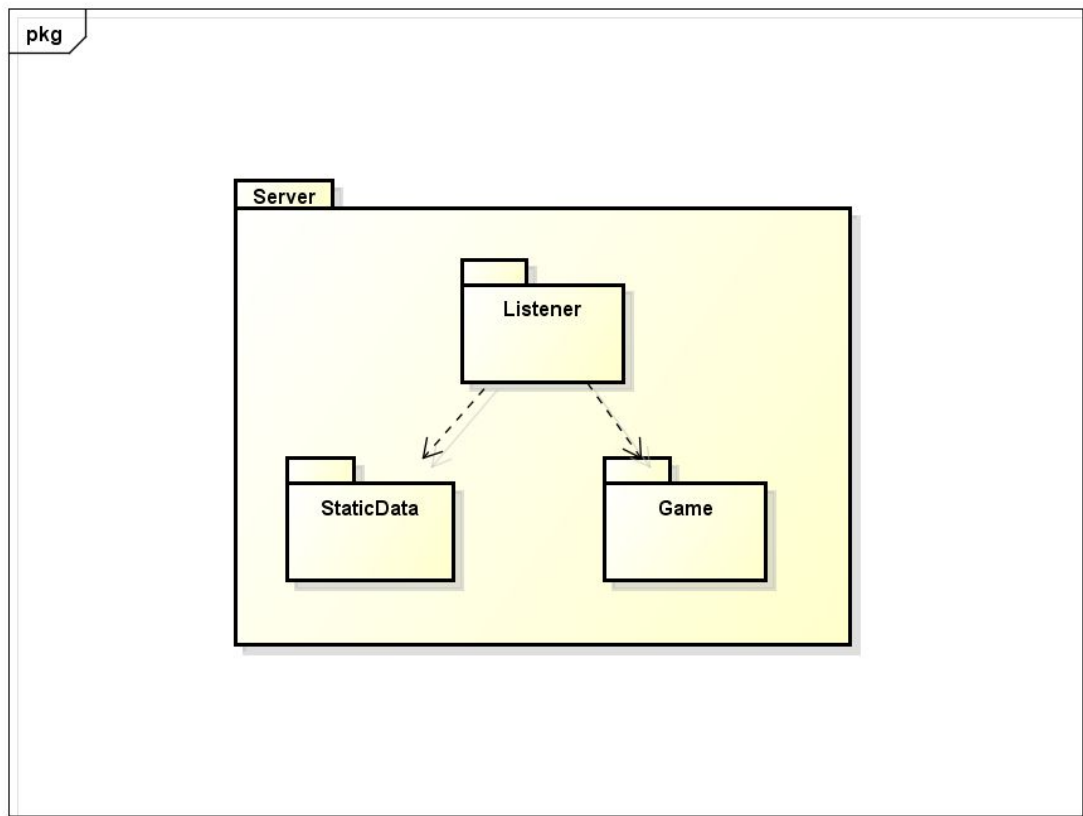


图 5 服务端架构设计

服务端系统架构中的对象分为 3 类：

- (1) Listener 对象，负责对与客户端的通信和客户端发来消息的监听与处理。
- (2) StaticData 对象，负责对服务器端所有用户统计数据保存和读写逻辑。
- (3) Game 对象，负责对正在进行的对战或协作游戏的处理。

服务端的组件接口：

接口 ID	连接组件	接口信息	
I1	连 接 Listener 与 StaticData	语法	Return (Response) Interface (Request)
		前置条件	对统计数据的操作请求正确
		后置条件	StaticData 处理对应的增、删、改、查 请求并响应
		不变量	操作请求信息
I2	连 接 Listener 与 Game	语法	Return (Response) Interface (Command)



		前置条件	无
		后置条件	对对应的操作进行确认并通知Listener转发此操作给进行同一游戏的其它玩家
		不变量	对游戏状态更改的请求

客户端的架构设计如下。

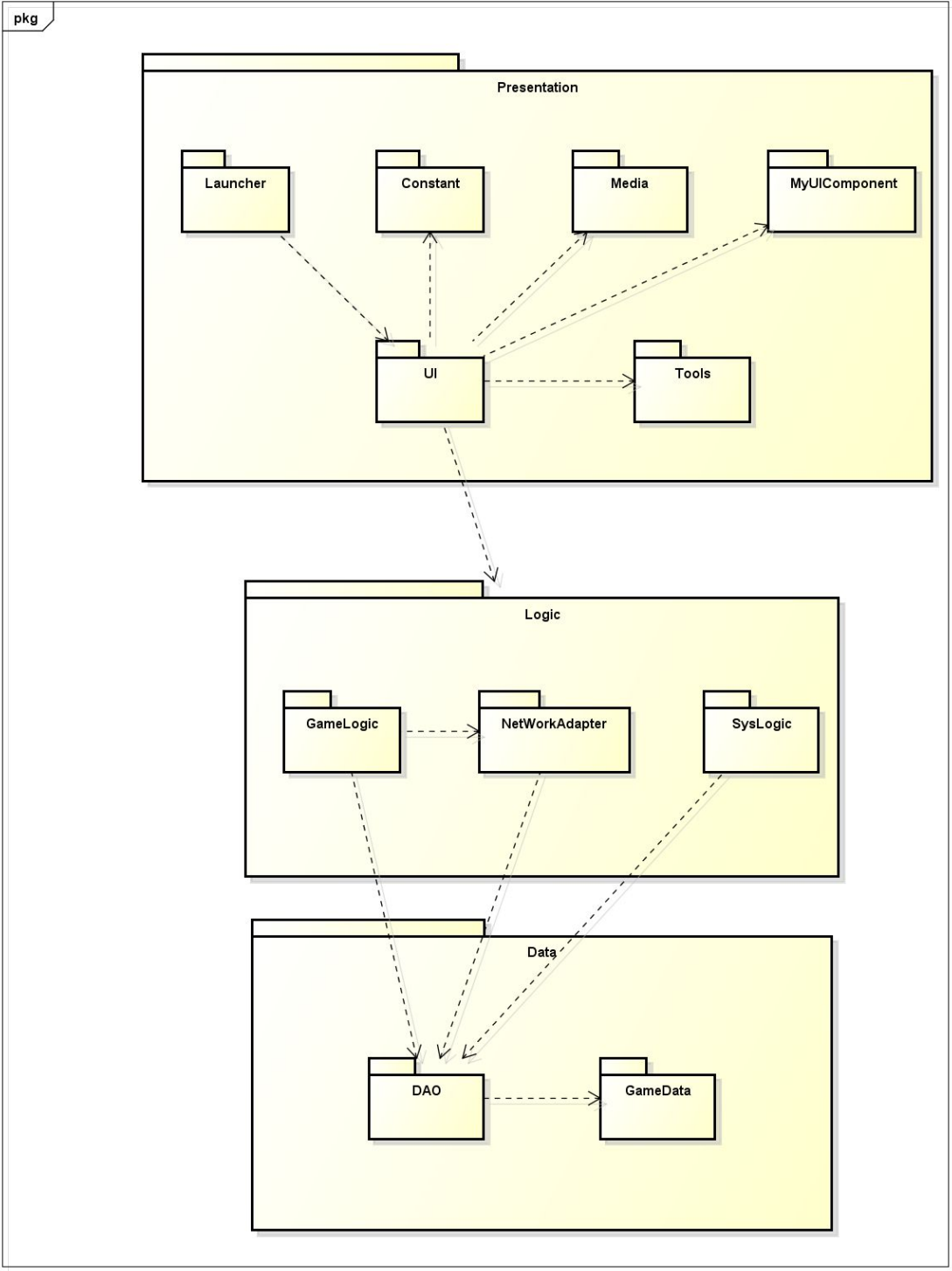


图 6 客户端架构设计

客户端系统架构中的对象分为 11 类：

- (1)Launcher 对象，负责启动游戏。
- (2)UI 对象，负责处理系统对展现和与用户的交互。
- (3)Constant 对象，负责处理游戏的不变参数。
- (4)Media 对象，负责对音乐等媒体文件的播放。
- (5)MyUIComponent 对象，负责各个界面控件的外观和事件响应。
- (6)Tools 对象，负责处理界面的特效。
- (7)GameLogic 对象，负责处理游戏进行中的逻辑判断。
- (8)SysLogic 对象，负责处理系统全局的逻辑判断。
- (9)NetworkAdapter 对象，负责处理协作/对战游戏中与服务器的网络通信。
- (10)DAO 对象，负责处理对本地游戏历史数据的操作。
- (11)GameData 对象，用于将游戏数据封装成数据实体。

客户端的组件和组件接口：

接口 ID	连接组件	接口信息	
I3	连接 Launcher 与 UI	语法	Return(Result) Interface()
		前置条件	无
		后置条件	游戏被尝试启动并返回结果
		不变量	无
I4	连接 UI 与 Constant	语法	Return(Result) Interface(Request)
		前置条件	请求获取的系统不变参数存在
		后置条件	返回队形的系统不变参数
		不变量	系统的所有不变参数
I5	连接 UI 与 Media	语法	Return(Result) Interface(Request)
		前置条件	请求对媒体文件进行的操作正确
		后置条件	执行请求对应的操作并返回结果
		不变量	媒体文件和操作媒体文件的请求
I6	连接 UI 与 MyUIComponent	语法	Return(Result) Interface()
		前置条件	无
		后置条件	返回所需的界面组件
		不变量	界面组件自身
I7	连接 UI 与 Tools	语法	Return(Result) Interface(Request)
		前置条件	无
		后置条件	执行请求对应的特效并返回结果
		不变量	特效的实际效果不因界面状态的变化而变化
I8	连接 UI 与 Logic	语法	Return(Result) Interface(Command)

		前置条件	无
		后置条件	相应的逻辑组件检查命令的正确性并执行合法的命令
		不变量	无
I9	连接 GameLogic 与 NetWorkAdapter	语法	Return(Result) Interface(Command)
		前置条件	无
		后置条件	网络组件无条件地将命令交付给服务器并返回交付结果
		不变量	命令信息
I10	连接 GameLogic 与 DAO	语法	Return(Result) Interface(NewData)
		前置条件	新的游戏历史数据正确
		后置条件	将新数据合并入历史数据，并返回操作结果。
		不变量	新产生的游戏数据
I11	连接 NetWorkAdapter 与 DAO	语法	Return(GameData) Interface()
		前置条件	无
		后置条件	DAO 返回当前的所有游戏数据中的统计数据部分
		不变量	本地保存的游戏历史数据
I12	连接 SysLogic 与 DAO	语法	Return(Result) Interface(Command)
		前置条件	对游戏数据的操作命令正确
		后置条件	DAO 对当前的游戏数据执行对应的操作并返回操作结果
		不变量	无
I13	连接 DAO 与 GameData	语法	Return(GameData) Interface(Command)
		前置条件	游戏历史数据存在
		后置条件	DAO 将新游戏数据与历史数据合并或从历史数据中取出需要的部分，封装后返回
		不变量	无