

天天爱消除

PC 版

# 详细设计文档

## ( 服务器 )

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. 5. 21	最初草稿，发布 至 SVN 供参考	V1.0

# 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 服务器模块的分解.....	5
2.2.1 服务器模块的职责 .....	6
2.2.2 服务器模块的接口规范 .....	6
2.2.3 启动模块的设计原理 .....	7
3.系统体系结构设计思路.....	7

# 1.引言

## 1.1 编写目的

本文档提供天天爱消除 PC 版的详细设计描述和模型图，在软件架构的基础上进行了进一步分析，为系统的维护、扩展和后续开发提供参考依据。

## 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工程师。

## 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
- (5) 天天爱消除 PC 版软件架构设计文档

# 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。

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。

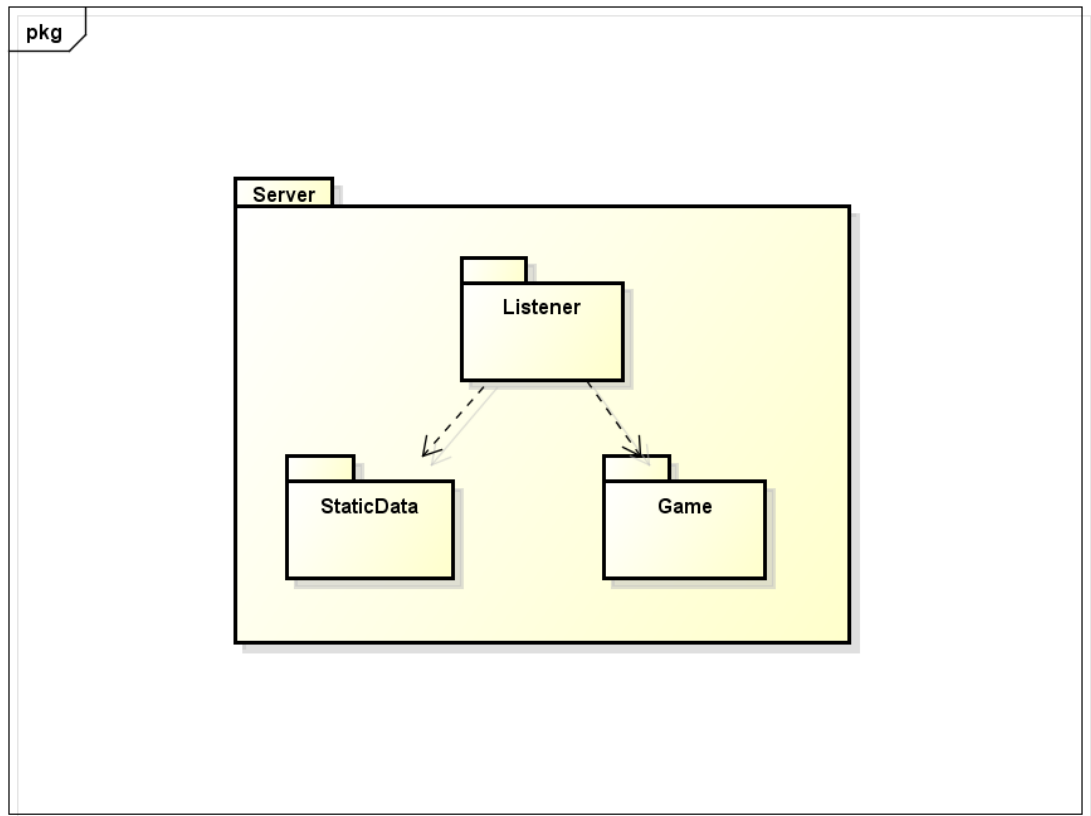


图 1 服务端的体系结构逻辑表达

## 2.2 服务器模块的分解

按照前期产生的《软件架构设计文档》，天天爱消除 PC 版的服务器模块详细设计类图如下。

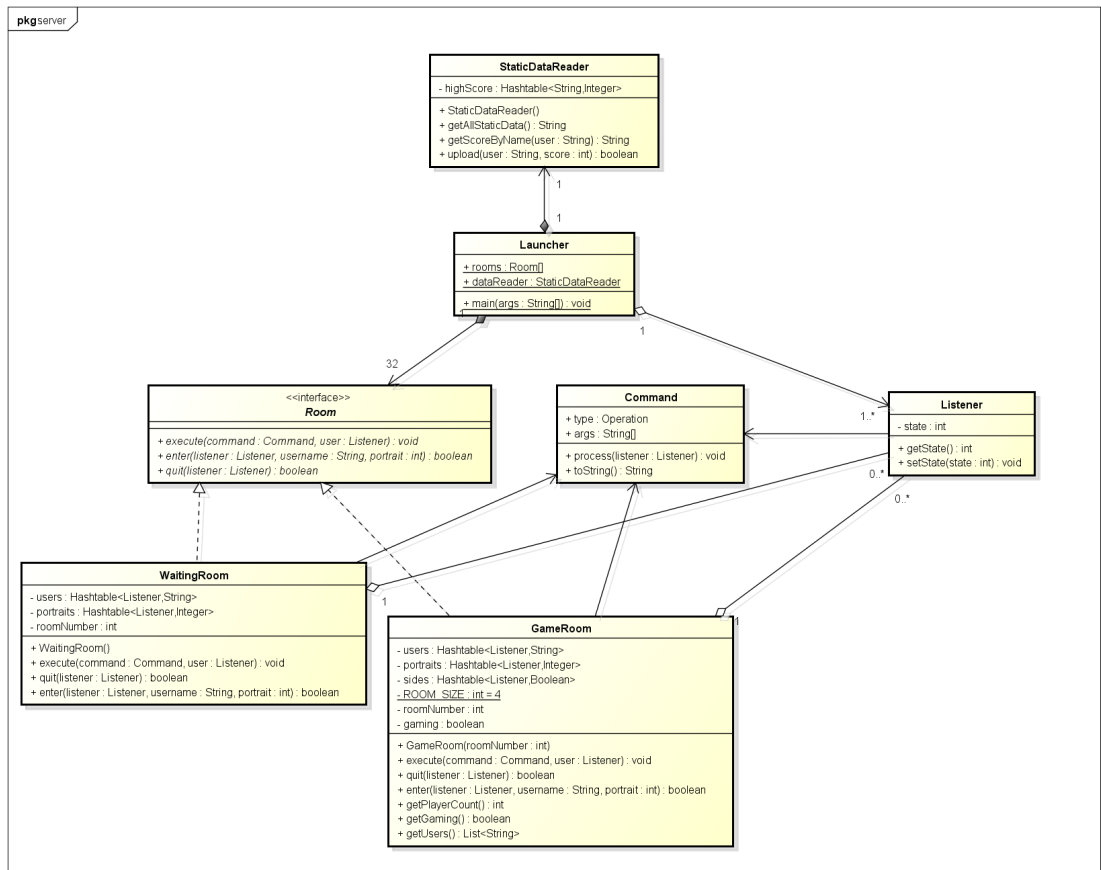


图 2 服务器模块详细类图

### 2.2.1 服务器模块的职责

类名称	所属层和模块	功能描述
Launcher	server	服务器启动器
StaticDataReader	server.staticdata	用于管理服务器端的统计信息，并对统计信息进行相应的读写与修改
WaitingRoom	server.game	实现游戏等待室内的相关功能
GameRoom	server.game	实现游戏房间内的相关功能
Command	server.listener	用于记录客户端传来命令的纯数据类
Listener	server.listener	通过 socket 与客户端进行交互

表 1 服务器模块中的类说明

### 2.2.2 服务器模块的接口规范

类名称	方法名称	功能描述
StaticDataReader	getAllStaticData()	获取服务器上记录的所有统计信息
	getScoreByName()	通过用户名获取指定用户的统计信息
	upload()	上传用户当次游戏的分数并与服务器已有记录合并
WaitingRoom	execute()	对相应玩家执行命令的相关操作
	quit()	玩家退出房间，进入等待室
	enter()	玩家进入游戏房间
GameRoom	execute()	对相应玩家执行命令的相关操作
	quit()	玩家离开等待室

	enter()	玩家进入等待室
	getPlayerCount()	获取当前游戏房间内的玩家人数
	getGaming()	获取当前游戏房间内是否在进行游戏
	getUsers()	获取当前游戏房间内的所有玩家用户名
Command	process()	用于记录客户端传来命令的纯数据类
Listener	getState()	获取当前用户的状态（所在房间）
	setState()	设定当前用户的状态（所在房间）

表 2 服务器模块中类的方法说明

2.2.3 启动模块的设计原理

启动模块 game.server.Launcher 是整个服务端的程序入口。服务端启动时，首先将一个等待室和数量预先设定好的游戏房间进行初始化，设为房间无人的空闲状态。之后生成全局唯一的 StaticDataReader 对象实例，之后所有有关统计数据的操作都透过此对象进行。随后开启一个 ServerSocket，在指定端口上监听来自客户端的连接请求。之后，除了进行相应的监听，其它操作都由由于新客户端连接服务器而生成的新线程来处理。

3.系统体系结构设计思路

服务器通过启动器开启一个端口监听来自客户端的连接，对于每个连接，服务器端开启一个新线程与之进行通信。连接到服务器的用户不是在等待室就是在游戏房间中，服务器通过为每一个线程标记一个状态来确定其所在的游戏场景。服务器获取来自客户端的每一条命令，经过一定的处理，与联机游戏相关的请求直接由命令对象调用服务器唯一的统计数据读写对象进行处理，其它命令由用户所在的房间根据用户和命令进行相应的处理，并返回给客户端相应的处理结果。如有必要，处理结果还将转发给其它与服务器连接的客户端。