**软件体系架构原型**

目录

[1 前言 1](#_Toc385188510)

[1.1 发布的日期和状态 1](#_Toc385188511)

[1.2 发布的组织机构 1](#_Toc385188512)

[1.3 作者 1](#_Toc385188513)

[1.4 变更历史 1](#_Toc385188514)

[2 总体介绍 1](#_Toc385188515)

[2.1 目的 1](#_Toc385188516)

[2.2 范围 1](#_Toc385188517)

[2.3 参考 2](#_Toc385188518)

[3 引用 2](#_Toc385188519)

[4 词汇表 2](#_Toc385188520)

[5 系统设计描述主体 2](#_Toc385188521)

[5.1 系统层次 2](#_Toc385188522)

[5.3 独立组件和组件接口 3](#_Toc385188523)

# 1 前言

## 1.1 发布的日期和状态

**发布日期：**2014-04-010

**状态：**描述软件体系架构原型

## 1.2 发布的组织机构

Mosaic Team @ Software Institute of Nanjing University.

## 1.3 作者

Rhett ([hjw12@software.nju.edu.cn](mailto:hjw12@software.nju.edu.cn)).

## 1.4 变更历史

|  |  |  |  |
| --- | --- | --- | --- |
| **版 本** | **作 者** | **版本描述** | **日 期** |
| V1.0 | Rhett | 发布到SVN服务器，提供给成员阅读并评审 | 2014-04-10 |

# 2 总体介绍

## 2.1 目的

本文档提供IceBreaker游戏的软件架构原型。

## 2.2 范围

本文档的读者是Mosaic Team团队内部的开发和管理人员，参考了[IEEE 1016-2009]中的架构描述推荐文档模版，用于指导下一循环的代码开发和测试工作。

## 2.3 参考

1. 《软件需求规格说明文档》，Mosaic Team;
2. 《系统设计文档模版》，[IEEE 1016-2009]；
3. 《软件架构设计文档》
4. 《计算与软件工程（卷三）：团队与软件开发实践》，骆斌，刘嘉，张瑾玉，黄蕾，2012。

# 3 引用

# 4 词汇表

|  |  |  |
| --- | --- | --- |
| **词 汇 名 称** | **词 汇 含 义** | **备 注** |
| IceBreaker | 产品游戏的名称 |  |
| View | MVC架构中的视图模块 |  |
| Controller | MVC架构中的控制器模块 |  |
| Model | MVC架构中的模型模块 |  |
| Dto | Data transfer object,数据传输对象 | 封装数据，传输 |
| Dao | Data access object,数据访问对象 | 直接对数据库进行读写 |
| Entity | 数据实体对象 |  |

# 5 系统设计描述主体

## 5.1 系统层次

系统层次结构如图1所示。



图1系统的层次结构

系统划分为以下3个逻辑层次。

1. 展示层：用于前台界面展示和配置的层次。
2. 业务层：包含业务控制和逻辑的层次。
3. 数据层：定义和存储系统中相关数据的层次。

## 5.3 独立组件和组件接口

客户端架构中的对象分为7类：

1. View对象，负责UI的展示和与用户的交互。
2. Viewservice对象，模型模块改变视图调用的服务。
3. Controller对象，负责获取用户的输入和改变模型。
4. Modelservice对象，控制器或服务器反馈信息改变模型调用的服务。
5. Model对象，封装游戏信息，响应视图的查询和控制器的改变。
6. Netservice对象，模型模块向服务器进行请求操作调用的服务。
7. Net对象，负责网络服务接口的实现模块，与服务器进行连接。

客户端的组件和组件接口：

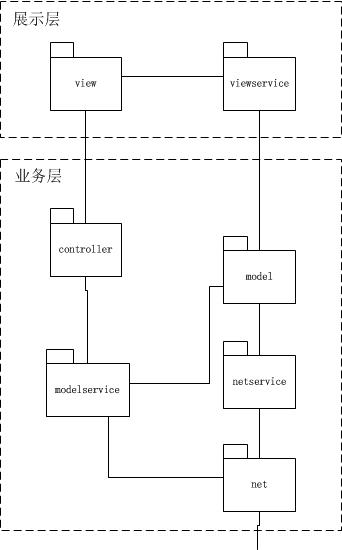


图3 客户端的组件

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | **连接组件** | **接口信息** | |
| I1 | 连接view和controller | 语法 | Return(Response)  Interface(Request) |
| 前置条件 | 用户的输入正确 |
| 后置条件 | 经过控制组件处理请求并且响应 |
| 不变量 | 用户请求信息 |
| I2 | 连接controller和modelservice | 语法 | Return()  Interface(Request) |
| 前置条件 | 无 |
| 后置条件 | 模型经过逻辑处理进行相应改变 |
| 不变量 | 无 |
| I3 | 连接model和viewservice | 语法 | Return()  Interface() |
| 前置条件 | 无 |
| 后置条件 | 视图响应模型进行改变 |
| 不变量 | 无 |
| I4 | 连接model和netservice | 语法 | Return()  Interface(Request) |
| 前置条件 | 与服务器正常连接 |
| 后置条件 | 服务器收到请求进行处理 |
| 不变量 | 模型状态 |
| I5 | 连接net和modelservice | 语法 | Return()  Interface() |
| 前置条件 | 无 |
| 后置条件 | 模型收到服务器反馈结果进行相应改变 |
| 不变量 | 无 |

服务器架构中的对象分为6类：

1. Net对象，负责与客户端的连接。
2. Service对象，调用服务器逻辑处理的服务。
3. Serviceimp对象，负责对于抽象接口的服务模块。
4. Dto对象，负责封装从dao获取的批量数据的接口。
5. Dao对象，负责与数据库实体交互，获取数据。
6. Entity对象，负责将数据库中获取的数据封装成数据尸体。

服务器端的组件和组件接口：

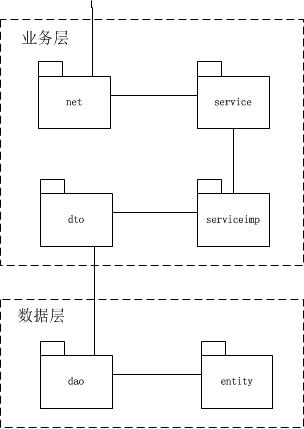


图4 服务器端的组件

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | **连接组件** | **接口信息** | |
| I1 | 连接net和service | 语法 | Return(result)  Interface() |
| 前置条件 | 无 |
| 后置条件 | Service进行逻辑处理并返回结果 |
| 不变量 | 无 |
| I2 | 连接service和dto | 语法 | Return(dataSet)  Interface(command) |
| 前置条件 | 无 |
| 后置条件 | 对应的dto组件调用特定dao类获取数据层数据，并返回数据集 |
| 不变量 | 无 |
| I3 | 连接dao和entity | 语法 | Return(data)  Interface(criteria) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | Dao中的类将entity对象写入数据库或从数据库中返回entity对象 |
| 不变量 | 无 |