# GameServer设计文档

此文档主要描述了整个GameServer（游戏服务器）的设计架构。便于开发人员能够迅速理解整个体系结构，建立大局观。

对于整个游戏服务器的功能，可以划分为两大部分。

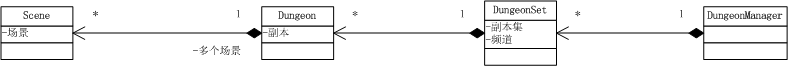
**游戏世界** 与 **玩法系统。**

**游戏世界**：多频道副本制的世界构成，场景及规则，场景管理与AOI(area of interest)系统，生物对象继承体系，生物对象的基础功能、状态机、AI、技能系统。

**玩法系统**：任务系统、VIP管理系统、装备系统等。

## 世界构成

整个游戏的所有场景和关卡由 DungeonManager管理。



DungeonManager管理多个DungeonSet。

DungeonSet。每个id的副本会对应一个DungeonSet。它管理该id的副本的所有频道。ChannelManager会协助DungeonSet管理频道。

Dungeon。每个副本实例对应于此类，它对应于某id副本的某个频道。

Scene。一个副本中可能会有多个场景。

可参考文档 <世界构成.vsd>

## 场景规则

每个Scene会对应有自己的Regular玩法规则。Regular表达该场景的规则相关逻辑，比如通关逻辑、开始结束规则等。



参考文档 <gameholder.vsd>

## 生物对象继承体系



场景中所有的对象，都是由如上图的继承体系派生出来。

参考文档 <gameholder.vsd>

## 生物对象基础功能



每个生物对象，都有对应的功能相关的模块，状态机、AI控制器、技能控制器、Buff控制器、行为控制器(移动、使用技能等接口，而不是直接调用BattleUnit)

## 各玩法系统

游戏中每个玩家会有相应的各种玩法系统，如任务、背包、装备、坐骑、宠物等。



在架构上会将每个玩法系统作为Player所拥有的一个对象来处理。

## 全局性规则和功能系统

**事件系统**

在游戏中发生各种逻辑抛出事件，需要响应该事件的模块监听该事件。

**游戏设置管理器 GameSettingManager**

对游戏中各种参数进行记录和配置。

**离线玩家数据管理器 OfflinePlayerManager**

管理、请求和保存。非在线玩家信息的管理器。