FreeEngine

一款开源的游戏服务端引擎，底层框架使用C++编写，核心概念少，易上手，学习成本低。整个架构基于模块机制。

注重模块的可复用性，降低功能逻辑耦合。脚本、网络、数据库等支持都是通过模块提供。 该引擎并不是为某一种网游类型设计，它更注重单个服务器架构设计，通过不同的模块组合，引擎适用于所有网络游戏类型。

该引擎设计目标为：

1. 简单、实用、易上手、方便、快捷、低耦合

2. 通过对已有模块的重组，快速构建服务器框架和功能逻辑

3. 提供丰富、稳定、高可复用的模块代码

**特性**

1. 跨平台，目前支持linux和windows系统；
2. 核心概念少，学习成本低，易上手；
3. 模块机制，所有特性可通过模块实现（网络、数据库、脚本支持等）；
4. 网络数据自动同步机制，网络操作对逻辑层透明，像写单机逻辑一样写服务器逻辑，同时底层网络模块只发送修改过的数据，最优化网络通信流量；
5. 数据库自动备份；
6. 所有游戏逻辑代码支持热更新，包括二进制逻辑和脚本逻辑。
7. 如果游戏逻辑使用c++编写，这里可以通过编码规范和静态检查屏蔽所有的指针操作，引擎内部通过标准异常捕获所有非法数据访问，可以解决所有崩溃问题和内存碎片问题。

**代码组织视图**



主要分3层结构

1. 跨平台api封装，兼容平台差异
2. 基础服务库，模块管理，启动文件，异常处理，常用的库和第三方库等
3. 模块代码，包括公共模块和游戏逻辑相关模块

**逻辑视图 通用卡牌休闲游戏**

通用结构，适用当前所有卡牌、休闲类游戏

主要工作量在蓝色部分的数据定义和游戏逻辑编写上，其它大部分由引擎提供

引擎原理，工作流程

1. 定义需要用到的数据
2. 用数据编译工具生成本地代码，（这里同时生成客户端和服务器代码，客户端需要根据需求改写生成代码，用于接收同步数据）
3. 自动生成的代码主要包括3部分，数据操作接口，脏数据筛选分类，数据格式化。数据操作接口导出给游戏逻辑使用，网络模块和数据库模块可以监视脏数据，如果检测到有数据修改，可以通过格式化输出，获取自己需要的数据。比如网络模块可以通过格式化输出把脏数据格式化成json格式，然后通过网络发往客户端。
4. 游戏逻辑是无状态的，所以可以随时进行热更，修复bug