MVC根据范围不同有两种理解

1. 界面范围，像XluaFramework，MVC组件都属于一个界面，同时创建和销毁。
2. 全局范围，这时候M用于整个游戏，不会随着界面销毁而销毁。

界面范围的写起来很麻烦，所以使用全局范围

MVC的对应关系：

M和V是多对多才合适，C辅助V进行工作，C和V是一对一。

MVC的职责：

M：主要的业务逻辑。

V：刷新界面的逻辑。

C：是最不好定义职责的，定义如下：

1. 胶水代码，一般情况下没有状态（属性），只有逻辑（方法），主要作为V修改M数据的中介。
2. 持有多个M对象，混合处理数据。
3. 分担V的任务，比如组织数据发送C2S请求消息。

MVC通信关系：

M：不持有V和C，通过事件刷新V。可以注册和派发事件。

V：持有M和C，对于M只读，如果是写M需要使用C。可以注册和派发事件。

C：持有M，协助V对M进行写操作。不能注册和派发事件。

收发消息：

发消息可以放在C层，收消息放在了NetHandler

也可以都放在Model