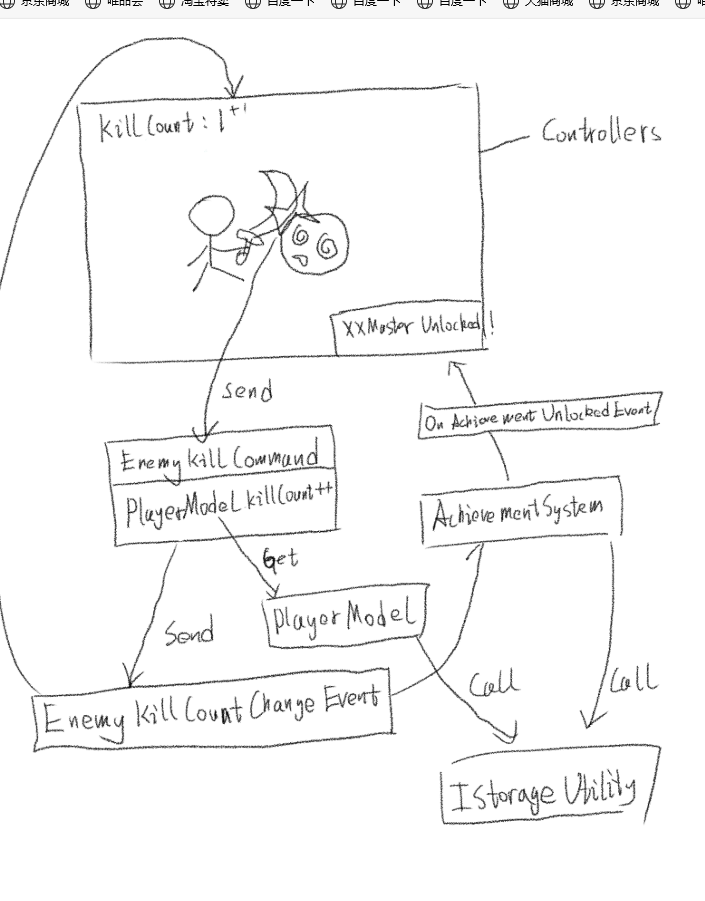
# QFamerwork

工具类从model获取数据，然后发送通知给Event，model和system在call 存储

事件再来更新UI的内容。



**QFramework系统设计架构分为四层及其规则：  
1、表现层：ViewController层。IController接口，负责接收输入和状态变化时的表现，一般情况下，MonoBehaviour均为表现层  
可以获取System  
可以获取Model  
可以发送Command  
可以监听Event  
2、系统层：System层。ISystem接口，帮助IController承担一部分逻辑，在多个表现层共享的逻辑，比如计时系统、商城系统、成就系统等  
可以获取System  
可以获取Model  
可以监听Event  
可以发送Event  
3、数据层：Model层。IModel接口，负责数据的定义、数据的增删查改方法的提供  
可以获取Utility  
可以发送Event  
4、工具层：Utility层。IUtility接口，负责提供基础设施，比如存储方法、序列化方法、网络连接方法、蓝牙方法、SDK、框架继承等。啥都干不了，可以集成第三方库，或者封装API  
5、除了四个层级，还有一个核心概念——Command  
可以获取System  
可以获取Model  
可以发送Event  
可以发送Command  
使用规则：  
1、IController更改ISystem、IModel的状态必须用Command  
2、ISystem、IModel状态发生变更后通知IController必须用事件或BindableProperty  
3、IController可以获取ISystem、IModel对象来进行数据查询  
4、ICommand不能有状态  
5、上层可以直接获取下层，下层不能获取上层对象  
6、下层向上层通信用事件  
7、上层向下层通信用方法调用（只是做查询，状态变更用Command），IController的交互逻辑为特别情况，只能用**