# 活动模块代码设计

我们所谓的活动，基本指的是，在一个界面，有一个tabList，可以滑动，点击任意一个图标或者按钮，然后界面的另一部分，显示一个组件，显示具体活动的数据结构。

我们游戏中的活动、福利、商城等，都是这样的一个活动框架中的内容，通过一些简单的小技巧，也可以设置单个活动内容，详情可以参考玩法界面的一些配置；

活动模块大体上分为两类，一个是数据层的框架和管理，一个是界面层的代码框架和管理，下面分别介绍；

## 数据层：

所有的活动数据都是BaseActivityData的子类，BaseActivityFactory是不同活动数据结构的工厂创建类，可以通过活动的type进行配置并自动解析活动；

每一个活动对应一个IActivityItemManager，这个是每一个活动的管理类基类，相当于每一个活动，都有一个Manager，对应包含这个ActivityData；

## 表现层：

所有的活动表现层，都继承自BaseActivityComponent，提供一些不同分页活动的数据和生命周期管理；

如果要添加一个大的分类，比如福利，界面本身可以参考CommonShopMallActivityView界面，都要继承自BaseBannerView去实现对应的功能；

## 数据表：

经历过上面的代码实现后，就可以通过配置表去配置对应的活动，涉及的表格如下：

1. ActivityItemConfig表：用来配置活动的一些数据，以及在tabList中的显示和周例中的显示信息；
2. ActivityConstructor表：配置不同的活动用不同的模板类（IActivityItemManager），以及相关依赖的客户端的数据；
3. ComponentExtensionConfig表：配置组件和FairyGUI的扩展类的关系，这个也可以写在代码里，但是这样就不方便热更新，用表格更加方便；
4. ActivityPointSourceConfig表：用来配置积分来源通用界面的一些逻辑；
5. ActivityRankViewConfig表：用来配置活动的排行榜数据，也可以用在别的地方；
6. RedTipNodeConfig表：红点表，配置树状结构的红点数据；

## 通用界面

ActivityRankView：游戏中通用的排行榜界面，支持多种Tab排列，具体看代码，支持通过活动获取数据，也可以自定义数据；

ActivityPointSourceView：游戏中通用的积分来源界面；

CommonHeroSelectView：通用的英雄选择界面；

ActivityTargetRewardPreviewView：活动目标预览的通用界面；

ActivityWeekCalendarView：周例界面；