**EOC项目程序快速入门**

1. 安装Unity，取客户端代码，设置tfs，流程参考 新人入职，工具安装及客户端克隆.docx

2. 运行游戏

* 打开eoc\_client工程
* ctrl+Q 打开launcher场景，启动游戏
* 体验游戏 使用GM命令参考 <http://10.24.0.24:8889/cmd>

3. 通过打断点熟悉游戏初始化流程

* 从Launcher场景的 GlobalClient 脚本开始，找到程序的入口点
* 熟悉GlobalClient的FSM机制，维护和切换游戏状态，GlobalManager的管理
* 熟悉游戏中几个主要Manager的作用和Init流程



4. 找到游戏的主逻辑 GameLogicManager 熟悉AppFacade:StartUp,断点到第一个UI界面加载 ViewBinder:Create



5. 熟悉游戏的核心MVC概念，游戏主要的逻辑都使用了MVC架构开发，在Assets\Scripts\Hotfix\Common\PureMVC





5. 了解主要代码的目录结构

Assets\Scripts\Client 客户端管理器，简单了解每种管理器的作用

Assets\Scripts\Client\Component 自定义组件如UI扩展

Assets\Scripts\Hotfix\MVC 逻辑部分

Assets\Scripts\Hotfix\MVC\Base 几个关键的MVC类 AppFacade,GameCmd, GameMediator, GameProxy

Assets\Scripts\Hotfix\MVC\CMD CmdConstant定义 Sproto服务器协议

Assets\Scripts\Hotfix\MVC\GlobalMediator **全局逻辑(主逻辑)**

Assets\Scripts\Hotfix\MVC\LuaBehaviour lua接口

Assets\Scripts\Hotfix\MVC\Troop 部队(寻路，排列，更新)

Assets\Scripts\Hotfix\MVC\ViewMediator **和界面相关的逻辑(主逻辑)**

Assets\Scripts\Hotfix\MVC\ViewMediator\Common UISubView通用界面逻辑

Assets\Skyunion\Runtime 重点关注UIManager的UI管理逻辑

6. 其他

策划数据的生成，目前策划数据在eoc的tfs中管理，策划\策划表格 下，用excel编辑，生成工具在 策划\策划表格\Config 下，运行convert.bat 生成csv和代码，在客户端中用C#解析数据，代码位置在 Assets\Scripts\Client\Config 下 XXXConfig.cs就是每个数据表的结构，用来解析表(参见DataService.cs), 数据的存储格式是.bin二进制格式，客户端位置在 Assets\BundleAssets\Config\Bin

UIId.cs UI界面的枚举，被UIManager管理

Sproto.cs 协议的定义

7. 逻辑功能划分

内城

大世界

PVE

UI相关，多语言

联盟

活动

领主

城市管理

聊天

新手引导

英雄

事件

美女

任务

剧情

情报

英雄抽奖

SDK，付费

分包打包机制

热更新

性能优化，美术规格制定

1. UI模块的制作流程
2. 制作预设



首先根据待制作的UI的级别选择工具中的选项来创建UI预设的公共模板。



以上是一级界面的模板预设



以上是二级界面的模板预设

自己的预设控件都在ContentView节点下制作。

制作后修改X级界面\_改名这个节点的名字后，拖放在Assets\BundleAssets\UI\目录下。

1. 生成模板代码

首先配置前端资源表格



上图为配置ClientResources.xlsm



上图为配置UI.xlsm

在编辑器的Project中找到制作并保存好的预设，点击鼠标右键。



Gen Prefab Code：生成View和Mediator代码，主要用于界面功能。

Gen SubView Code：生成view和Mediator代码，主要用于界面内嵌套的功能，例：item

Gen Prefab Code Single View：生成View代码不生成Mediator

1. 代码编写

生成的模板代码一般分为View和Mediator两种

逻辑写在Mediator类中

例：

initData:初始化数据方法

BindUIEvent：绑定事件方法

BindUIData：绑定数据方法

Update：每帧调用方法，注：应用Update方法需提前开启，IsOpenUpdate = true

WinClose:关闭界面回调方法

OpenAniEnd：打开界面动画播放完毕回调方法

WinFocus：界面至于焦点回调方法

1. 其他

1 事件系统

使用事件系统的时候，使用EventManager的相关方法，例：

EventManager.AddEventListener<long,int,BattleUIType>(CmdConstant.ShowWorldDamageText, OnShowDamageText);

老的事件逐渐被废除，例：

AppFacade.GetInstance().SendNotification(CmdConstant.FightUpdateHeroLevel, captainLevelUpData);

计时器功能的应用，例：



参数说明

duration:延迟多少秒调用

onComplete:回调方法

onUpdate:循环方法

isLooped：是否循环

UseRealTime：是否使用真实时间

AutoDestroyOwner：是否自动销毁

2 查表方法：

CoreUtils.dataService.QueryRecord<ConfigDefine>(0).vitalityReduceUnit

ConfigDefine为表名

0为id

vitalityReduceUnit为字段

3 打开UI方法：

CoreUtils.uiManager.ShowUI(UIID.UI\_Win\_GiftLimitedTime,null,packageInfo);

UIID.UI\_Win\_GiftLimitedTime打开UI的枚举

null是打开回调方法

packageInfo打开界面传递的参数

4 关闭UI方法：

CoreUtils.uiManager.CloseUI(UIID.UI\_Pop\_ArmySelect);

UIID.UI\_Pop\_ArmySelect关闭界面的UI枚举