# AlphaDiagram 简称AD

## 当前版本 0.4.3

主要分为以下几个部分，请通过本文件夹中其他部分获取部分帮助，代码端更多功能请自行阅览代码

当前AD中有大量代码位于AD.Experimental命名空间中，通常是一些并未完全测试的功能，不过通过实践后也暂时不会离开这个命名空间

* **AD.BASE 抽象框架 衍生自QFrameWork** 
  + **IBase – IBaseMap 双类转换接口**
  + **ADException AD异常**
  + **AD架构 核心架构接口以及标准类的基本实现**
  + **AD Invokeable Call 委托 监听 事件 回调**
  + **Property绑定实体 由扩展函数支持的资源封装类**
  + **ObjectExtension 主要是类型转换相关内容**
  + **Event System Hander 事件触发器**
  + **DebugExtenion 调试以及Log**
  + **Process 进程（实验性）**
* **ADUI UI组件及附属资源等**
* **AD.Core 核心组件**
* **Extension 扩展函数&Utility**

# ADGlobalSystem

以下简称ADG

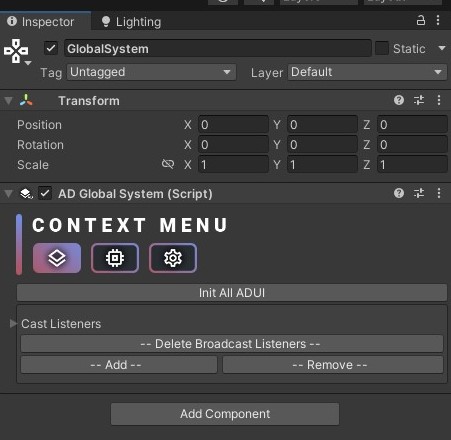
ADG以单例模式存在，并且可以通过instance静态成员访问该实例

ADG的实现深度高于大部分AD组件和实现，浅于全部AD.BASE内容

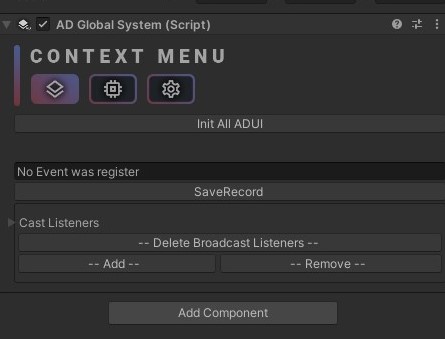
## Inspector

* **Content部分**

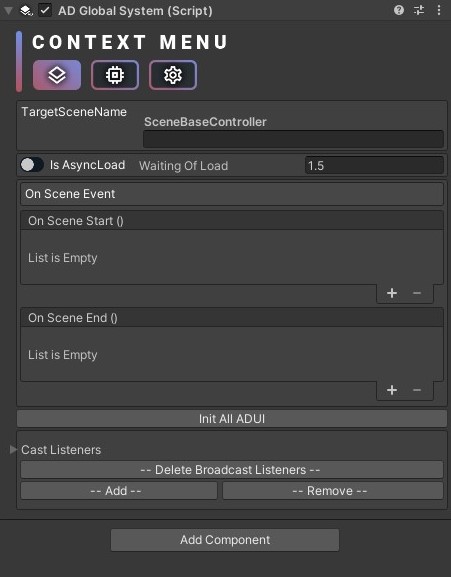
非运行时非场景管理器模式



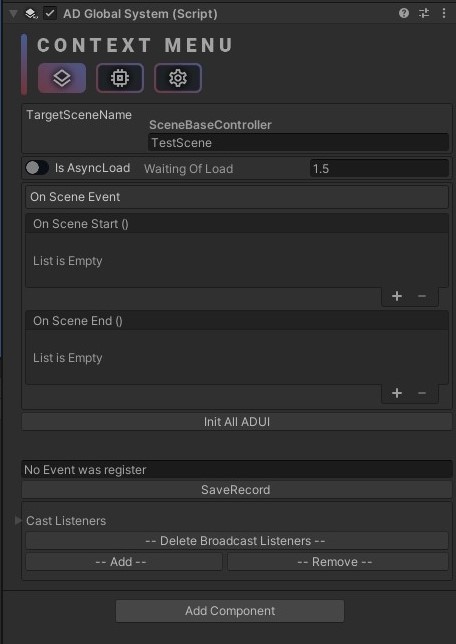
运行时非场景管理器模式



非运行时场景管理器模式



运行时场景管理器模式



ADG可以通过Init All ADUI按钮来尝试初始化所有支持初始化的ADUI组件（包括注册的IADUI）

ADG的Cast Listeners列表为

public ADSerializableDictionary<string, List<MonoBehaviour>> CastListeners

其功能由Broadcast模块提供。使用该模块功能请调用public HashSet<object> SendMessage(string key, params object[] args)

运行时ADG会显示出当前注册的按键，并且可以通过SaveRecord立即保存由ADG UtilityRecord模块收集的信息

当ADG被设为场景管理器时，可以设置目标场景名，加载场景时是否异步，最少等待多少秒；若不设置TargetSceneName则会默认设置为当前场景名称，若设置为常量public const string \_BackSceneTargetSceneName = "\_BACK\_"则会尝试返回前一个场景（需要也由ADG进行场景跳转）；OnSceneStart与OnSceneEnd分别会在场景加载与场景跳转时触发

ADG（SceneBaseController）的场景跳转函数为public override void OnEnd()

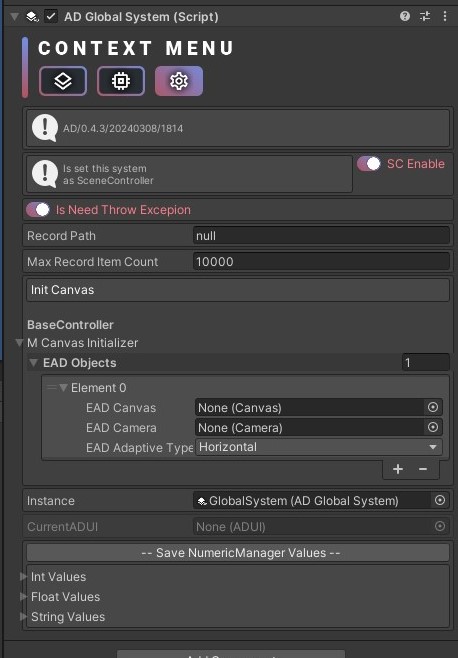
* **Resources**

图形用户界面, 文本

描述已自动生成

为本场景AD应用的默认ADUI等组件，如右键create menu时生成的物体均由此处Prefab克隆

* **Setting**

****

显示ADG保存的版本号以及版本时间

是否设置为场景管理器

是否在生成错误消息时依然抛出实际Excepion

由ADG保存的消息log保存位置，若为null则保存至默认位置

最大消息限制，保存的消息log总数不会超过该值，并且先进先出

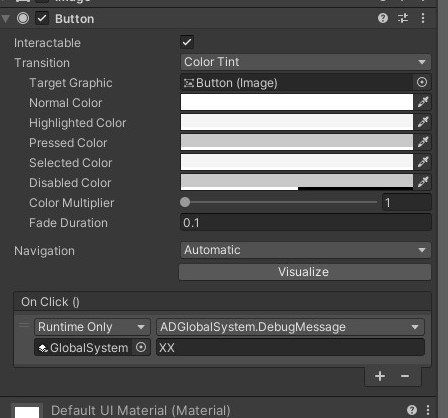
BaseController内容：Canvas初始化，在场景初始化时将目标World Canvas的大小根据目标Camera以AdaptiveType指定的模式进行自适应

单例指示器Instance

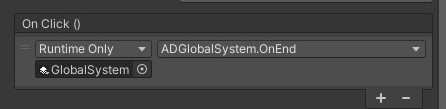
CurrentADUI会指示当前哪个ADUI（最晚）进入了Select状态，若没有，则为None

数值管理器部分，Save按钮可以手动保存当前的数值管理器内容，数值管理器为字典键值对

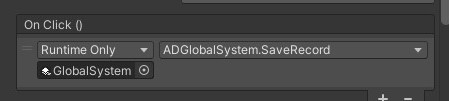
## ADG上那些工具函数



Debug消息发送



触发场景跳转



保存消息log