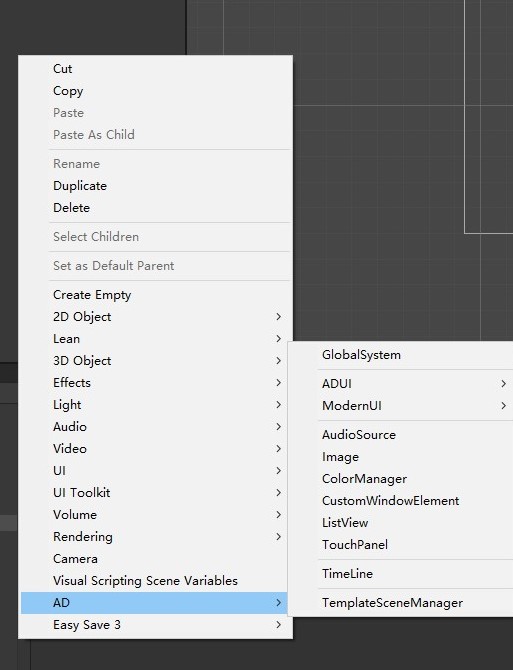
# 注意，使用前需要先生产ADGlobalSystem



# 你可以通过右键来快速的进行生成

# ADUI

## AD/Derivation/Object.Core/ADUI/ADUI

接口 public interface ICanInitializeBehaviourContext

* Void InitializeContext() 初始化外置事件环境

接口 public interface IADUI : ICanInitializeBehaviourContext

* IADUI Obtain(int serialNumber) 搜索目标IADUI实例
* IADUI Obtain(string elementName) 搜索目标IADUI实例
* String ElementName.get 获取IADUI实例的元素名
* String ElementName.set 设置IADUI实例的元素名
* Int SerialNumber 获取IADUI实例的序列号
* Bool IsNeedContext.get 是否需要在初始化时使用外置的事件环境
* BehaviourContext Context.get 获取依赖的事件环境

类 public abstract class ADUI : MonoBehaviour, IADUI

* **继承ADUI的子类禁止在其自身的Awake消息函数中使用ADUI的业务逻辑**
* 常量成员 DefaultNumericManagerName 当数值管理器的匹配名称为此值时不尝试初始化与保存
* 静态成员 ADUI CurrentSelect.get 获取当前光标指向的ADUI实例（假如其允许）
* 静态成员 List<IADUI> Items.get 获取全部注册的IADUI（假如其允许）
* 静态成员 int TotalSerialNumber.get 获取最大序列号
* 静态成员 string UIArea 获取当前光标指向ADUI实例的元素区域名称（假如其允许）
* 私有成员 BehaviourContext \_Context 事件环境
* 公共成员 BehaviourContext Context.get 获取事件环境
* 公共成员 bool Select 是否被光标指向（假如其允许）
* 公共成员 string ElementName 元素名
* 公共成员 int SerialNumber 序列号
* 公共成员 string ElementArea 元素区域名称
* 静态函数 public static void Initialize(IADUI obj) 注册并初始化IADUI实例
* 静态函数 public static void Initialize(IADUI obj, string numericManagerName) 注册并初始化该IADUI实例，若实现了INumericManager接口则进行数值管理器初始化（调用SetupByNumericManager）
* 静态函数 public static void DestroyADUI(IADUI obj) 销毁实例时应当调用该函数以取消注册状态
* 静态函数 public static ADOrderlyEvent<PointerEventData> InitializeContextSingleEvent(ADOrderlyEvent<PointerEventData> Event, params UnityAction<PointerEventData>[] calls) 协助事件环境中监听的注册
* 静态函数 public static ADOrderlyEvent<BaseEventData> InitializeContextSingleEvent(ADOrderlyEvent<BaseEventData> Event, params UnityAction<BaseEventData>[] calls) 协助事件环境中监听的注册
* 静态函数 public static ADOrderlyEvent<AxisEventData> InitializeContextSingleEvent(ADOrderlyEvent<AxisEventData> Event, params UnityAction<AxisEventData>[] calls) 协助事件环境中监听的注册
* 静态函数 public static void SetValue\_NumericManagerName(string NumericManagerName, float value) 保存数值管理器中的值到ADGlobalSystem中
* 静态函数 public static void SetValue\_NumericManagerName(string NumericManagerName, int value) 保存数值管理器中的值到ADGlobalSystem中
* 静态函数 public static void SetValue\_NumericManagerName(string NumericManagerName, string value) 保存数值管理器中的值到ADGlobalSystem中
* 静态函数 public static void GetValue\_NumericManagerName(string NumericManagerName, out float value) 从ADGlobalSystem中获取值
* 静态函数 public static void GetValue\_NumericManagerName(string NumericManagerName, out int value) 从ADGlobalSystem中获取值
* 静态函数 public static void GetValue\_NumericManagerName(string NumericManagerName, out string value) 从ADGlobalSystem中获取值
* 工具函数 public virtual void TurnsActive(GameObject target) 默认为设置该物体是否处于活动状态
* 工具函数 public virtual void OnPointerEnter(PointerEventData eventData) 默认为ADUI的区域检测
* 工具函数 public virtual void OnPointerExit(PointerEventData eventData) 默认为ADUI的区域检测
* 工具函数 public virtual void InitializeContext() 初始化事件环境
* 受保护的函数 protected virtual void HowSetupByNumericManager()
* 公共函数 public void SetupByNumericManager(string numericManagerName) 当参数不以” Default”字符串开始时，执行HowSetupByNumericManager
* 公共函数 public virtual IADUI Obtain(int serialNumber) 获取该序列号且完成注册的ADUI实例
* 公共函数 public virtual IADUI Obtain(string elementName) 获取该元素名称且完成注册的ADUI实例
* 公共函数 public virtual IADUI TryObtain(int serialNumber) Obtain(int serialNumber)的不报错版本
* 公共函数 public virtual IADUI TryObtain(string elementName) Obtain(string elementName)的不报错版本
* 公共函数 public virtual List<IADUI> ObtainAll(Predicate<IADUI> \_Right) 根据谓词搜索所有符合要求的ADUI实例

接口 public interface IButton : IADUI

* IButton SetTitle(string title); 设置按钮上的文本
* IButton AddListener(UnityAction action); 添加监听
* IButton RemoveListener(UnityAction action); 移除监听
* IButton RemoveAllListeners(); 移除全部监听

接口 public interface IBoolButton : IADUI

* bool isOn.get 获取布尔按钮的值
* bool isOn.set 设置布尔按钮的值
* IBoolButton AddListener(UnityAction<bool> action); 添加监听
* IBoolButton RemoveListener(UnityAction<bool> action); 移除监听

接口 public interface IDropdown : IADUI

* void AddOption(params string[] texts); 添加选项
* void RemoveOption(params string[] texts); 移除选项
* void ClearOptions(); 清理选项
* void Select(string option); 选中某项
* void AddListener(UnityAction<string> action); 添加监听
* void RemoveListener(UnityAction<string> action); 移除监听

接口 public interface IInputField : IADUI

* TMP\_InputField source.get 源组件
* string text.get 获取当前的文本值
* string text.set 设置当前的文本值
* InputFieldValueProperty ValueProperty.get 获取一个当前输入所转换的浮点值的绑定器
* InputFieldProperty TextProperty.get 获取一个当前文本输入的绑定器
* BindPropertyJustSet<string, BindInputFieldAsset> Input.get 获取一个只输入的字符串绑定器
* BindPropertyJustGet<string, BindInputFieldAsset> Output.get 获取一个只读的字符串绑定器
* void AddListener(UnityAction<string> action, PressType type = PressType.OnEnd); 添加指定类型监听
* void RemoveAllListener(PressType type = PressType.OnEnd); 移除指定类型全部监听
* void RemoveListener(UnityAction<string> action, PressType type = PressType.OnEnd); 移除指定类型监听
* void SetPlaceholderText(string text); 设置空文本时的填充
* IInputField SetText(string text); 设置文本
* IInputField SetTextWithoutNotify(string text); 设置文本并且不触发任何监听

枚举 public enum PressType

* OnSelect
* OnEnd

接口 public interface INumericManager

* void SetupByNumericManager(string numericManagerName);

接口 public interface INumericManager<T> : INumericManager

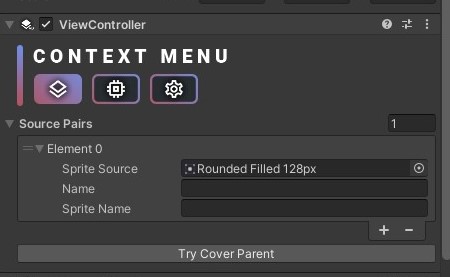
* void NumericManager(T value);

# 以上即为ADUI的标准内容

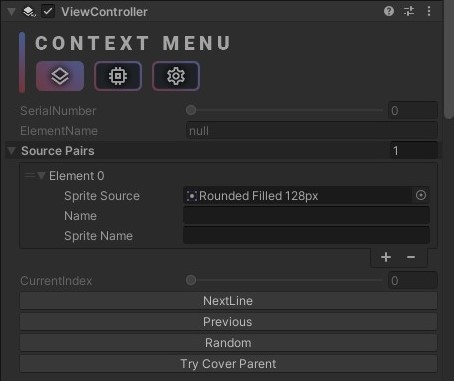
## AD/Derivation/Object.Core/View & Audio/ViewController

该组件用于管理一个Image组件 **密封类**

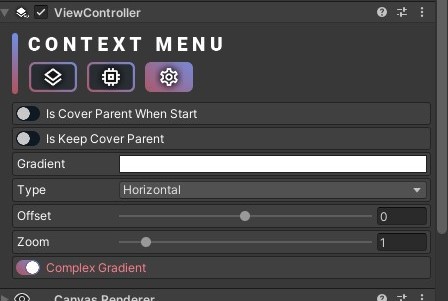
非运行时



运行时



设置



ViewController的源组件是Image，并将当前使用的SourcePair设为显示的图像，可以将Next等函数作为监听制作一个图像流

TryCoverParent按钮按下时会立刻尝试覆盖父物体，通常用于不随分辨率变化时布设背景

IsCoverParentWhenStart会指示是否应该在物体Start时保持原分辨率并覆盖父物体，通常用于布设背景

IsKeepCoverParent会指示当前每一帧是否都进行一次覆盖操作，通常用于分辨率动态变化时布设背景，性能开销较大

Gradient，Type，Offset，Zoom，Complex Gradient会共同控制对图像的渐变颜色效果

通过TimeLine或动画机等控制手段可以实现大量动画效果

代码端

public ViewController SetTransparentChannelCollisionThreshold(float value)

设置透明通道阈值

public ViewController SetMaterial(Material material)

设置材质

SourcePair系列函数及public ViewController Refresh()

控制流，立即刷新当前状态

SetColor系列函数

设置颜色

public ViewController BakeAudioWaveformFormAudioCilp(AudioClip clip)

以默认设置烘焙一张波形图并立即显示在当前位置上（不覆盖CurrentPair）

public void SetupCoverParent()

覆盖父物体

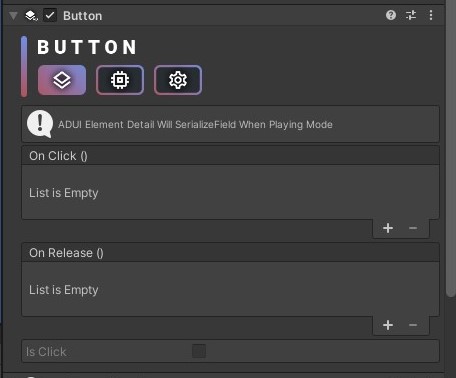
Load系列函数

从Resources，Url或是本地路径中加载贴图（可指示立即替换Current或是添加在流的末尾）

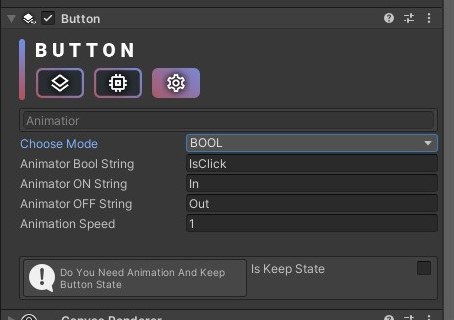
## Assets/AD/Derivation/Object.Core/ADUI/Button

该组件实现了一个简单的按钮

非运行使用动画机时







如果你不想让按钮能够维持按下的状态，便取消IsKeepState，这同样会使得所有动画不再播放

选择ButtonAnimatorMode以指示动画机如何运作，配置三项String以支持自定义的动画机