# IMCUI API 文档 0.0.1

## IMCUI 基于 UNITY 5.3 UGUI 源码扩展而来，对于UGUI 原生并且没有修改过的API不再解释（请自己参考unity 官方api 文档）

## IMCUI API文档不会对所有private类型的变量&方法&属性&枚举&对象等进行注解

## IMCUI 并非对UGUI源码的所有类(Class)名全部更改为以IMC开头！仅限所有的控件类以及个别类

## IMCUI 将UGUI中的所有类全部合并到命名空间(namespace) IMCUI.UI下

# 目录

[public enum ControlType 1](#_Toc23318)

[public enum ContainerType 1](#_Toc19141)

[IMCUIBehaviour (基类) 1](#_Toc13662)

[OnDestroy() 1](#_Toc2003)

[m\_controlType 1](#_Toc23595)

[controlType 1](#_Toc13487)

[m\_containerType 1](#_Toc16492)

[containerType 2](#_Toc23432)

[parent 2](#_Toc5292)

[blocker 2](#_Toc14727)

[initialize 2](#_Toc19215)

[rectTransform 2](#_Toc859)

[customID 2](#_Toc3133)

[intanceID 2](#_Toc13045)

[active 3](#_Toc13985)

[raycast 3](#_Toc20558)

[graphics 3](#_Toc12456)

[alpha 3](#_Toc29995)

[interact 3](#_Toc28126)

[Initialize() 3](#_Toc3135)

[UnInit(int time = 0) 3](#_Toc17954)

[GetContainerObject() 4](#_Toc3608)

[T GetBaseType<T>() 4](#_Toc30533)

[GetSiblintIndex() 4](#_Toc21180)

[SetSiblingIndex(int index) 4](#_Toc11472)

[SetAsFirstSibling() 4](#_Toc1247)

[SetAsLastSibling() 4](#_Toc1729)

[Unparent() 4](#_Toc27672)

[TweenCallBack 5](#_Toc8073)

[MoveTo(Vector3 position, float time, UnityAction callBack = null) 5](#_Toc11655)

[RotationTo(Vector3 angle, float time, UnityAction callBack = null) 5](#_Toc21129)

[ScaleTo(Vector3 size, float time, UnityAction callBack = null) 5](#_Toc24254)

[AlphaTo(float targetAlpha, float time, UnityAction callBack = null) 5](#_Toc20446)

[anchoredPosition3D 5](#_Toc25844)

[anchoredPosition 6](#_Toc12101)

[anchorMax 6](#_Toc10950)

[anchorMin 6](#_Toc855)

[offsetMax 6](#_Toc18710)

[offsetMin 6](#_Toc14194)

[pivot 6](#_Toc25542)

[rect 6](#_Toc5632)

[sizeDelta 7](#_Toc16736)

[GetLocalCorners(Vector3[] fourCornersArray) 7](#_Toc519)

[GetWorldCorners(Vector3[] fourCornersArray) 7](#_Toc7403)

[SetInsetAndSizeFromParentEdge(RectTransform.Edge edge, float inset, float size) 7](#_Toc1572)

[SetSizeWithCurrentAnchors(RectTransform.Axis axis, float size) 7](#_Toc27516)

[UIMirror(bool isMirror) 7](#_Toc11353)

[IMCText 8](#_Toc23368)

[Awake() 8](#_Toc27993)

[alpha 8](#_Toc16445)

[interact 8](#_Toc15164)

[raycast 8](#_Toc621)

[cachedTextGenerator 8](#_Toc2748)

[IMCImage 9](#_Toc1237)

[SetNativeSize() 9](#_Toc28476)

[Awake() 9](#_Toc17022)

[alpha 9](#_Toc23624)

[interact 9](#_Toc10538)

[raycast 9](#_Toc13560)

[SaveImageToPath(string path) 9](#_Toc13984)

[IMCRawImage 10](#_Toc3360)

[Awake() 10](#_Toc22313)

[alpha 10](#_Toc12447)

[interact 10](#_Toc23802)

[raycast 10](#_Toc20323)

[IMCButton 11](#_Toc27820)

[Text 11](#_Toc26234)

[Awake() 11](#_Toc13954)

[alpha 11](#_Toc14018)

[interact 11](#_Toc10985)

[raycast 11](#_Toc24050)

[Initialize() 11](#_Toc3019)

[AddListener(UnityAction action) 12](#_Toc3347)

[RemoveListener(UnityAction action) 12](#_Toc2481)

[RemoveAllListener() 12](#_Toc28581)

[StartWithinStipulatedTimeCall(float time, UnityAction action) 12](#_Toc21196)

[StopWithinStipulatedTimeCall() 12](#_Toc12545)

[StartOutsideStipulatedTimeCall(float time, UnityAction<GameObject> action) 12](#_Toc27475)

[StopOutsideStipulatedTimeCall() 12](#_Toc14777)

[AddOnPointerLongListener(UnityAction<GameObject> action) 13](#_Toc8675)

[RemoveOnPointerLongListener(UnityAction<GameObject> action) 13](#_Toc19714)

[StartDelayCall(float time, UnityAction action) 13](#_Toc18366)

[StopDelayCall() 13](#_Toc11888)

[onPointerLongPress 13](#_Toc21530)

[longpressTime 13](#_Toc31370)

[LongPressEnable 14](#_Toc4837)

[isZoomAnimation 14](#_Toc28925)

[isPlayAudio 14](#_Toc17064)

[clip 14](#_Toc9599)

[Create(string buttonTitle = "Button", Sprite backgroundPicture = null) 14](#_Toc25192)

[Create(UnityAction call, string buttonTitle = "Button") 14](#_Toc29847)

[Create(UnityAction call, string buttonTitle = "Button", Sprite backgroundPicture = null) 14](#_Toc14267)

[OnPointerEnter(PointerEventData eventData) 15](#_Toc17388)

[OnPointerExit(PointerEventData eventData) 15](#_Toc5744)

[IMCToggle 16](#_Toc4555)

[Awake() 16](#_Toc32372)

[alpha 16](#_Toc13817)

[interact 16](#_Toc2522)

[raycast 16](#_Toc1558)

[Initialize() 16](#_Toc13363)

[label 16](#_Toc25115)

[AddListener(UnityAction<bool> action) 16](#_Toc3481)

[RemoveListener(UnityAction<bool> action) 17](#_Toc4531)

[RemoveAllListener() 17](#_Toc16150)

[isPlayAudio 17](#_Toc24968)

[clip 17](#_Toc13141)

[IMCSlider 18](#_Toc10162)

[value 18](#_Toc3516)

[Awake() 18](#_Toc20718)

[alpha 18](#_Toc11096)

[interact 18](#_Toc25674)

[raycast 18](#_Toc1103)

[Initialize() 18](#_Toc6500)

[IMCScrollbar 19](#_Toc14763)

[Awake() 19](#_Toc15728)

[alpha 19](#_Toc15041)

[interact 19](#_Toc17825)

[raycast 19](#_Toc1660)

[Initialize() 19](#_Toc8616)

[IMCDropdown 20](#_Toc28757)

[OptionData 20](#_Toc14187)

[OptionData.itemToggle 20](#_Toc19907)

[OptionData.itemLabel 20](#_Toc31004)

[OptionData.itemBackground 20](#_Toc22626)

[OptionData.itemCheckmark 20](#_Toc20819)

[ShowItemAction 20](#_Toc12838)

[HideItemAction 20](#_Toc15595)

[onValueChanged 21](#_Toc9300)

[onHideChanged 21](#_Toc26784)

[onSelect 21](#_Toc14105)

[isShow 21](#_Toc6339)

[alpha 21](#_Toc28403)

[interact 21](#_Toc5001)

[raycast 22](#_Toc9801)

[IMCInputField 23](#_Toc19122)

[Awake() 23](#_Toc27880)

[alpha 23](#_Toc10151)

[interact 23](#_Toc25659)

[raycast 23](#_Toc13625)

[Initialize() 23](#_Toc31913)

[IMCCanvas 24](#_Toc30768)

[Name 24](#_Toc22435)

[buttonClickClip 24](#_Toc16036)

[forms 24](#_Toc18250)

[PlayAudio(AudioClip clip) 24](#_Toc19277)

[RegisterForm(IMCForm form) 24](#_Toc8340)

[UnRegisterForm(IMCForm form) 24](#_Toc2875)

[FindFormByName(string name) 24](#_Toc8320)

[FindFormsByName(string name) 25](#_Toc25336)

[FindFormByByte<T>() where T:IMCForm 25](#_Toc20197)

[FindFormsByByte<T>() where T : IMCForm 25](#_Toc654)

[RaycastSwitch(bool isSwitch) 25](#_Toc3318)

[IMCMessageBox 26](#_Toc6789)

[MessageBoxButtonAttribute 26](#_Toc21679)

[background 26](#_Toc541)

[title 26](#_Toc6333)

[content 26](#_Toc8210)

[buttonParent 26](#_Toc28838)

[closeButton 26](#_Toc12960)

[closeObject 26](#_Toc23359)

[btns 27](#_Toc14511)

[Awake() 27](#_Toc29097)

[alpha 27](#_Toc3401)

[interact 27](#_Toc30580)

[raycast 27](#_Toc4035)

[Deploy(UnityAction<int> choiceBack, string messageBoxTitle, string messageBoxContent, Sprite messageBoxBackground, MessageBoxButtonAttribute[] messageBoxAttribute) 27](#_Toc22679)

[Deploy(UnityAction<int> choiceBack) 28](#_Toc17993)

[Deploy(UnityAction<int> choiceBack, string messageBoxTitle, string messageBoxContent) 28](#_Toc27736)

[Deploy(UnityAction<int> choiceBack, string messageBoxTitle, string messageBoxContent, Sprite messageBoxBackground) 28](#_Toc5547)

[SetButton(int buttonIndex, string buttonTitle = "", Sprite buttonSprite = null, UnityAction buttonAction = null) 28](#_Toc19555)

[SetButton(int buttonIndex, MessageBoxButtonAttribute buttonAttribute) 28](#_Toc16533)

[SetButton(IMCButton buttonObject, string buttonTitle = "", Sprite buttonSprite = null, UnityAction buttonAction = null) 29](#_Toc18844)

[RemovButtonElement(int buttonIndex) 29](#_Toc12707)

[RemoveButtonElement(IMCButton buttonObject) 29](#_Toc5227)

[IMCCascadeDropdown 30](#_Toc28394)

[dropDowns 30](#_Toc7323)

[Awake() 30](#_Toc19178)

[alpha 30](#_Toc3333)

[interact 30](#_Toc17934)

[raycast 30](#_Toc10641)

[GetDropDownsIndex(IMCDropdown dropDown) 30](#_Toc27937)

[currentDropdown 31](#_Toc19035)

[IMCSwitchButton 32](#_Toc9306)

[Awake() 32](#_Toc17671)

[alpha 32](#_Toc3060)

[interact 32](#_Toc23930)

[raycast 32](#_Toc21109)

[content 32](#_Toc24978)

[onClick 32](#_Toc10749)

[moveSpeed 33](#_Toc354)

[isOn 33](#_Toc9661)

[onContent 33](#_Toc26341)

[offContent 33](#_Toc10223)

[isContent 33](#_Toc27069)

[isColor 33](#_Toc7748)

[onColor 33](#_Toc14541)

[offColor 34](#_Toc3849)

[canvasGroup 34](#_Toc28737)

[IsOnToggle(bool ison) 34](#_Toc24707)

[IMCOrderShowButtons 35](#_Toc21740)

[targets 35](#_Toc25944)

[leftBtn 35](#_Toc29582)

[rightBtn 35](#_Toc12589)

[Awake() 35](#_Toc2937)

[alpha 35](#_Toc8010)

[interact 35](#_Toc10025)

[raycast 36](#_Toc833)

[LeftBtnClick() 36](#_Toc6859)

[RightBtnClick() 36](#_Toc31723)

[ResetTargets(int index = 0) 36](#_Toc8133)

[IMCHyperlinksLabel 37](#_Toc25402)

[onHrefClick 37](#_Toc15807)

[hyperlinksLaberFontSize 37](#_Toc27059)

[hyperlinksLaberColor 37](#_Toc29889)

[SetHyperlinksContent(string url, string content) 37](#_Toc8839)

[SetHyperlinksContent(string url, string content, Color contentColor, int contentFontSize) 37](#_Toc31066)

[SetHyperlinksContent(string url, string content, string contentColor, int contentFontSize) 37](#_Toc11682)

[IMCProgressBar 39](#_Toc6897)

[Awake() 39](#_Toc22330)

[animationStyle 39](#_Toc28079)

[value 39](#_Toc144)

[sprite 39](#_Toc23713)

[background 39](#_Toc1028)

[tempView 39](#_Toc11779)

[backgroundImage 40](#_Toc5822)

[rotateSpeed 40](#_Toc4823)

[isTempViewCustomSize 40](#_Toc31841)

[tempViewSize 40](#_Toc25648)

[isTempViewPosition 40](#_Toc17840)

[tempViewPosition 40](#_Toc985)

[Begin(Sprite tempViewSprite = null, Sprite backgroundSprite = null) 40](#_Toc22800)

[Begin(AnimationStyle style, Sprite tempViewSprite = null, Sprite backgroundSprite = null) 41](#_Toc17223)

[Stop(bool isDestroy=false) 41](#_Toc15423)

[IMCUIContainer 42](#_Toc2799)

[canvasGroup 42](#_Toc4294)

[image 42](#_Toc3449)

[raycast 42](#_Toc234)

[alpha 42](#_Toc28010)

[interact 42](#_Toc2339)

[Initialize() 42](#_Toc12600)

[GetContainerObject() 43](#_Toc10861)

[AddControl(IMCUIBehaviour control) 43](#_Toc229)

[AddControl(IMCUIBehaviour control, Vector3 pos) 43](#_Toc13285)

[AddControls(List<IMCUIBehaviour> list) 43](#_Toc16793)

[AddControls(List<IMCUIBehaviour> list, List<Vector2> posList) 43](#_Toc1685)

[RemoveControl(IMCUIBehaviour control) 43](#_Toc1739)

[RemoveControlAndDestroy(IMCUIBehaviour control) 43](#_Toc3164)

[RemoveAllControl() 44](#_Toc19965)

[RemoveAllControlAndDestroy() 44](#_Toc9042)

[FindControl<T>() 44](#_Toc24858)

[FindControlByCustomID(string customID) 44](#_Toc2334)

[FindControlByCustomID<T>(string customID) 44](#_Toc23792)

[FindControlByName(string name) 44](#_Toc28174)

[FindControlByName<T>(string name) 45](#_Toc1126)

[FindControlByNameAndCustomID(string name, string customID) 45](#_Toc22011)

[FindControlByNameAndCustomID<T>(string name, string customID) 45](#_Toc3665)

[FindControls<T>() 45](#_Toc23869)

[FindControlsByCustomID(string customID) 45](#_Toc19232)

[FindControlsByCustomID<T>(string customID) 45](#_Toc27187)

[FindControlsByName(string name) 46](#_Toc10420)

[FindControlsByName<T>(string name) 46](#_Toc22581)

[FindControlsByNameAndCustomID(string name, string customID) 46](#_Toc20199)

[FindControlsByNameAndCustomID<T>(string name, string customID) 46](#_Toc5636)

[FindControlByIntanceID(int intanceID) 46](#_Toc20800)

[InitializeChildControls() 46](#_Toc9793)

[GetChildControls() 47](#_Toc10702)

[CanvasGroupSwitch(bool m\_switch) 47](#_Toc4973)

[MoveHouse(IMCUIBehaviour control, IMCUIContainer form, IMCUIContainer to, Vector2 pos, int tabViewIndex = 0) 47](#_Toc17904)

[IMCForm 48](#_Toc15284)

[canvas 48](#_Toc24617)

[showOnAwake 48](#_Toc6264)

[showTargetPosition 48](#_Toc6579)

[closeTargetPosition 48](#_Toc1528)

[isBlockerToggle 48](#_Toc26105)

[isDontDestroy 48](#_Toc14712)

[blockerShowStyle 49](#_Toc32423)

[InitializeAction 49](#_Toc24961)

[ShowAction 49](#_Toc3439)

[CloseAction 49](#_Toc2342)

[UnInitAction 49](#_Toc24752)

[blockerCallBack 49](#_Toc24182)

[showCallBack 49](#_Toc27300)

[closeCallBack 50](#_Toc15752)

[Show() 50](#_Toc1735)

[Show(Vector3 showTargetPosition, bool blockerToggle = false, IMCBlocker.ShowStyleEnum showStyle = IMCBlocker.ShowStyleEnum.Transparent, UnityEvent showCallBack = null, UnityAction blockerCallBack = null) 50](#_Toc11448)

[Close() 50](#_Toc8074)

[Close(Vector3 closeTargetPosition, UnityEvent closeCallBack = null) 50](#_Toc7191)

[UnInit(int time = 0) 51](#_Toc24213)

[Awake() 51](#_Toc2335)

[Initialize() 51](#_Toc9251)

[OnDestroy() 51](#_Toc28960)

[IMCScrollRect 52](#_Toc9792)

[parentScrollRect 52](#_Toc4103)

[Awake() 52](#_Toc24304)

[InitializeChildControls() 52](#_Toc5739)

[Initialize() 52](#_Toc26757)

[IMCTabView 53](#_Toc6411)

[toggles 53](#_Toc8621)

[contents 53](#_Toc19739)

[Awake() 53](#_Toc32519)

[OnEnable() 53](#_Toc16)

[TabViewControlShowOrClose(bool nuclearBombSwitch, IMCTabViewToggle tvt) 53](#_Toc24123)

[AddControl(int index, IMCUIBehaviour control) 53](#_Toc28373)

[AddControl(int index, IMCUIBehaviour control, Vector3 pos) 54](#_Toc3814)

[public void AddControl(int index, IMCUIBehaviour control, Vector3 pos) 54](#_Toc32646)

[AddControl(IMCUIBehaviour content, IMCUIBehaviour control) 54](#_Toc10206)

[AddControl(IMCUIBehaviour content, IMCUIBehaviour control, Vector2 pos) 54](#_Toc14884)

[InitializeChildControls() 54](#_Toc13204)

[GetCurrentOperationContentGameObjectIndex() 54](#_Toc32746)

[SetShowContent(int index) 54](#_Toc21672)

[IMCGroup 55](#_Toc15900)

[Awake() 55](#_Toc19428)

[IMCBlocker 56](#_Toc143)

[Instance 56](#_Toc21301)

[Is\_blockerPointerClick 56](#_Toc10558)

[ShowStyleEnum 56](#_Toc7524)

[Create(List<IMCUIBehaviour> targetUis, ShowStyleEnum showStyle, UnityAction callback = null, bool isDestroy = true, Transform parent = null) 56](#_Toc32159)

[Awake() 56](#_Toc2201)

[OnDestroy() 57](#_Toc32668)

[RemoveStack(IMCUIBehaviour removeTarget) 57](#_Toc16877)

[Screenshot() 57](#_Toc30556)

[IMCDebug 58](#_Toc19095)

[Instance 58](#_Toc31806)

[debugDatas 58](#_Toc4984)

[Log(string content, GameObject go = null) 58](#_Toc32428)

[LogError(string content, GameObject go = null) 58](#_Toc18964)

[LogWarning(string content, GameObject go = null) 58](#_Toc8643)

[Show() 58](#_Toc2611)

[Close() 59](#_Toc27349)

[Clear() 59](#_Toc458)

[AddFontSize() 59](#_Toc15947)

[ReduceFontSize() 59](#_Toc16597)

[tabView 59](#_Toc7301)

[IMCDragMove 60](#_Toc32065)

[panelRectTransform 60](#_Toc26261)

[rangeLimit 60](#_Toc4910)

[IMCUIAnimation 61](#_Toc29205)

[MoveTo(IMCUIBehaviour obj, Vector3 targetPos, float time, UnityAction callback = null) 61](#_Toc3067)

[RotationTo(IMCUIBehaviour obj, Vector3 targetAngle, float time, UnityAction callbacke = null) 61](#_Toc11365)

[ScaleTo(IMCUIBehaviour obj, Vector3 targetScale, float time, UnityAction callback = null) 61](#_Toc2818)

[AlphaTo(IMCUIBehaviour obj, float alpha,float time,UnityAction callback=null) 61](#_Toc20750)

[IMCUIResouceManager 62](#_Toc13551)

[LoadControl<T>(string path, Transform parent = null) 62](#_Toc5161)

[LoadControl(string path, string name, Transform parent = null) 62](#_Toc3863)

[LoadControl(string pathAndName, Transform parent = null) 62](#_Toc26861)

## public enum ControlType

public enum ControlType

控件类型枚举，会在所有控件的Awake函数执行时赋值

用途：可以判断该对象的控件类型

## public enum ContainerType

public enum ContainerType

容器类型枚举，会在所有控件的Awake函数执行时赋值

用途：可以判断该对象的容器类型

## IMCUIBehaviour (基类)

public abstract class IMCUIBehaviour : MonoBehaviour

### OnDestroy()

protected virtual void OnDestroy()

即将销毁函数

在控件销毁之前调用，并且判断父级是否为容器类型，将自身从父级的容器controls数组中将其移除

### m\_controlType

protected ControlType m\_controlType;

控件类型枚举变量

用途：继承此类可以设置该控件的控件类型

### controlType

public ControlType controlType

控件类型枚举属性【此属性只有get方法】

用途：方便外部调用该属性进行判断/归类

### m\_containerType

protected ContainerType m\_containerType;

容器类型枚举变量

用途：继承此类可以设置该控件的容器类型

### containerType

public ContainerType containerType

容器类型枚举属性【此属性只有get方法】

用途：方便外部调用该属性进行判断/归类

### parent

public IMCUIBehaviour parent

父级属性【此属性与transform.parent并非一个，此属性的类型为IMCUIBehaviour类型】

用途：容器类 （IMCUIContainer）用于设置非容器类控件设置容器类空间为父级

### blocker

public IMCBlocker blocker

阻拦器属性

用途：指定该控件的对应的阻拦器控件

### initialize

protected bool initialize = false;

是否初始化变量？

用途：控制控件禁止多次初始化方法调用

### rectTransform

public RectTransform rectTransform

矩形变换属性【此属性只有get方法】

用途：可以对UI控件进行缩放/位移/设置锚点/设置pivot

### customID

public string customID;

自定义ID变量

用途：可以设置指定的 customID，用于判断当前的控件是否为目标控件

### intanceID

public int intanceID

系unity自带API，只不过在该类中添加，方便使用【该属性只有get方法】

用途：instanceID是每一个Object独有的，用这个instanceID可以准确的找到指定的Object

### active

public bool active

激活属性

用途：用此属性可以设置/获取控件的激活状态

### raycast

public virtual bool raycast

射线属性

用途：用此属性可以设置控件射线的开关状态【对于多层级UI，会遍历子物体】

### graphics

protected Graphic[] graphics

图形数组属性【此属性只有get方法】

用途：？？？

### alpha

public virtual float alpha

α 透明度属性【该属性为虚属性】

用途：重写此属性自定义属性效果

### interact

public virtual bool interact

交互属性【该属性为虚属性】

用途：重写此属性自定义属性效果

### Initialize()

public virtual void Initialize()

初始化函数，在由IMCCanvas控件Awake函数触发自身以下所有UI的初始化函数【该函数为虚函数】

用途：重写此函数自定义初始化效果【一般控件都已经重写此函数】

### UnInit(int time = 0)

public virtual void UnInit(int time = 0)

卸载函数，销毁控件，参数time 变量默认为0即立即销毁，若大于0则延迟销毁【该函数为虚函数】

用途：重写次函数自定义效果

### GetContainerObject()

public virtual IMCUIContainer GetContainerObject()

获得容器对象函数【该函数为虚函数】

用途：？？？

### T GetBaseType<T>()

public virtual T GetBaseType<T>()

返回组件的类型函数【该函数为虚函数】

用途：？？？

### GetSiblintIndex()

public virtual int GetSiblintIndex()

获取组件当前Hierarchy面板中的层级函数【该函数为虚函数】

用途：用于获取层级判断层级关系

### SetSiblingIndex(int index)

public virtual void SetSiblingIndex(int index)

设置组件当前Hierarchy面板中的层级函数【该函数为虚函数】

用途：用于设置层级

### SetAsFirstSibling()

public virtual void SetAsFirstSibling()

将组件移动至当前Hierarchy面板层级最上方【该函数为虚函数】

用途:用于设置层级

### SetAsLastSibling()

public virtual void SetAsLastSibling()

将组件移动至当前Hierarchy面板层级最下方【该函数为虚函数】

用途：用于设置层级

### Unparent()

public IMCUIBehaviour Unparent()

解除父子关系函数，并返回组件

用途:解除父子关系

### TweenCallBack

public UnityAction TweenCallBack;

渐变回调？委托

用途：执行MoveTo RotationTo ScaleTo AlphaTo函数时回调

### MoveTo(Vector3 position, float time, UnityAction callBack = null)

public void MoveTo(Vector3 position, float time, UnityAction callBack = null)

位移动画函数，参数position为目标地点，time为动画时间，callBack为执行完回调【起始位置是以当前位置为准】

用途：位移操作

### RotationTo(Vector3 angle, float time, UnityAction callBack = null)

public void RotationTo(Vector3 angle, float time, UnityAction callBack = null)

旋转动画函数，参数angle为目标角度，time为动画时间，callBack为执行完回调【起始角度是以当前角度为准】

用途：旋转操作

### ScaleTo(Vector3 size, float time, UnityAction callBack = null)

public void ScaleTo(Vector3 size, float time, UnityAction callBack = null)

缩放动画函数，参数size为目标缩放值，time为动画时间，callBack为执行完回调【起始scale值是以当前大小(size)值为准】

用途：缩放操作

### AlphaTo(float targetAlpha, float time, UnityAction callBack = null)

public virtual void AlphaTo(float targetAlpha, float time, UnityAction callBack = null)

阿尔法动画函数（透明度动画函数），参数targetAlpha为目标阿尔法值【0~1之间】，time为动画时间，callBack为执行完回调【起始阿尔法值为当前阿尔法值为准】【该函数为虚函数】

用途：渐显渐隐操作

### anchoredPosition3D

public Vector3 anchoredPosition3D

锚点3D位置属性

用途：设置自身在UI画布（Canvas）上的3D位置

### anchoredPosition

public Vector2 anchoredPosition

锚点2D位置属性

用途：设置自身在UI画布（Canvas）上的2D位置

### anchorMax

public Vector2 anchorMax

锚点最大值属性

用途：更改锚点

### anchorMin

public Vector2 anchorMin

锚点最小值属性

用途：更改锚点

### offsetMax

public Vector2 offsetMax

最大偏移量属性

用途：用于计算UI相对右上角锚点的偏移

### offsetMin

public Vector2 offsetMin

最小偏移量属性

用途：用于计算UI相对做小脚锚点的偏移

### pivot

public Vector2 pivot

中心点属性

用途：用于设置/获取中心点（小蓝圈）的位置

### rect

public Rect rect

矩阵属性

用途：可以设置/获取当前UI的矩阵信息（x,y,width,height）

### sizeDelta

public Vector2 sizeDelta

大小增量属性

用途：可以设置/获取当前UI的大小

### GetLocalCorners(Vector3[] fourCornersArray)

public void GetLocalCorners(Vector3[] fourCornersArray)

获取折角的本地坐标函数【获取矩阵的四个点位置】

用途：？？

### GetWorldCorners(Vector3[] fourCornersArray)

public void GetWorldCorners(Vector3[] fourCornersArray)

获取折角的世界坐标函数【获取矩阵的四个点位置】

用途：？？

### SetInsetAndSizeFromParentEdge(RectTransform.Edge edge, float inset, float size)

SetInsetAndSizeFromParentEdge(RectTransform.Edge edge, float inset, float size)

设置从父变换边缘插入和尺寸函数【edge对齐方式，inset距离边缘距离，size宽度】

用途：用途设置UI的位置

### SetSizeWithCurrentAnchors(RectTransform.Axis axis, float size)

public void SetSizeWithCurrentAnchors(RectTransform.Axis axis, float size)

设置当前锚点大小函数【axis指定轴，size宽度】

用途:用于设置UI的大小

### UIMirror(bool isMirror)

public void UIMirror(bool isMirror)

镜像UI函数【isMirror为flase为镜像ui】【影响子集】

用途：用于设置UI镜像翻转显示

## IMCText

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### alpha

public override float alpha

透明度属性【取值范围0~1】【该属性为虚属性】

用途：更改组件的透明度

### interact

public override bool interact

交互开关属性【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【该属性为虚属性】

用途：判断该控件是否需要射线检测

### cachedTextGenerator

public TextGenerator cachedTextGenerator

缓存文本生成器属性

用途：可以获取很多该控件有用的属性，比如文字符号数量【换行也算一个字符】、顶点数量、控件大小、行数、字号等等

## IMCImage

### SetNativeSize()

public override void SetNativeSize()

设置本地大小函数【该函数为虚函数】

用途：可以根据sprite大小将控件也缩放至对应大小

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### alpha

public override float alpha

透明度属性【取值范围0~1】【该属性为虚属性】

用途：更改组件的透明度

### interact

public override bool interact

交互开关属性【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【该属性为虚属性】

用途：判断该控件是否需要射线检测

### SaveImageToPath(string path)

public bool SaveImageToPath(string path)

保存Image的path图片到指定路径函数【依赖于FileSystem脚本】

用途：可以保存Image的Sprite图片到指定路径下

## IMCRawImage

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### alpha

public override float alpha

透明度属性【取值范围0~1】【该属性为虚属性】

用途：更改组件的透明度

### interact

public override bool interact

交互开关属性【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【该属性为虚属性】

用途：判断该控件是否需要射线检测

## IMCButton

### Text

public IMCText Text

对button子物体text控件的索引属性【该属性为只读属性】

用途：获取text控件，对其控件进行操作

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### alpha

public override float alpha

透明度属性【取值范围0~1】【该属性为虚属性】

用途：更改组件的透明度

### interact

public override bool interact

交互开关属性【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【该属性为虚属性】

用途：判断该控件是否需要射线检测

### Initialize()

public override void Initialize()

重写初始化函数，使其只能进行一次初始化操作【该函数为虚函数】

用途：需要初始化的代码在重写该函数后添加

### AddListener(UnityAction action)

public void AddListener(UnityAction action)

添加侦听委托函数，参数action为无参委托

用途：动态添加委托

### RemoveListener(UnityAction action)

public void RemoveListener(UnityAction action)

删除侦听函数，参数action为需要删除的委托

用途：动态删除委托

### RemoveAllListener()

public void RemoveAllListener()

删除全部侦听函数

用途：动态清除所有的委托

### StartWithinStipulatedTimeCall(float time, UnityAction action)

public void StartWithinStipulatedTimeCall(float time, UnityAction action)

按住抬起在规定时间内回调函数，参数time为时间，action为需要绑定的无参回调【执行完一次后自动清空回调】

用途：用于特定情况下调用

### StopWithinStipulatedTimeCall()

public void StopWithinStipulatedTimeCall()

关闭StartWithinStipulatedTimeCall函数所执行的操作

用途：用于停止StartWithinStipulatedTimeCall的绑定

### StartOutsideStipulatedTimeCall(float time, UnityAction<GameObject> action)

public void StartOutsideStipulatedTimeCall(float time, UnityAction<GameObject> action)

按住抬起在规定时间后回调函数，参数time为时间，action为有参回调【回调参数为按钮自身的gameobject】【执行完一次后自动清空回调】

用途：用于特定情况下调用

### StopOutsideStipulatedTimeCall()

public void StopOutsideStipulatedTimeCall()

关闭StartOutsideStipulatedTimeCall函数所执行的操作

用途：用于停止StartOutsideStipulatedTimeCall的绑定

### AddOnPointerLongListener(UnityAction<GameObject> action)

public void AddOnPointerLongListener(UnityAction<GameObject> action)

添加长按委托侦听事件函数，参数action为需要添加的委托

用途：动态添加长按事件委托

### RemoveOnPointerLongListener(UnityAction<GameObject> action)

public void RemoveOnPointerLongListener(UnityAction<GameObject> action)

删除长按委托侦听事件函数，参数action为需要删除的委托

用途：动态删除长按事件委托

### StartDelayCall(float time, UnityAction action)

public void StartDelayCall(float time, UnityAction action)

延迟执行函数，按下后延迟time秒后回调，参数time为时间，action为无参回调【执行完一次后自动清空回调】

用途：用途特定情况下调用

### StopDelayCall()

public void StopDelayCall()

关闭StartDelayCall函数所执行的操作

用途：用于停止StartDelayCall的绑定

### onPointerLongPress

public UnityAction<GameObject> onPointerLongPress;

StartOutsideStipulatedTimeCall函数所绑定委托

用途：更改onPointerLongPress回调委托

### longpressTime

public float longpressTime;

长按时间变量

用途：更改长按时间

### LongPressEnable

public bool LongPressEnable = false;

长按事件开关变量

用途：更改长按事件是否触发开关

### isZoomAnimation

public bool isZoomAnimation

是否缩放动画属性

用途：更改按钮是否需要缩放动画

### isPlayAudio

public bool isPlayAudio

是否播放音频属性

用途：更改该属性确定当前按钮是否有按钮点击音效

### clip

public AudioClip clip

按钮点击音效属性

用途：添加该音效则按钮触发当前音效否则触发IMCCanvas组件下的“统一”按钮音效

### Create(string buttonTitle = "Button", Sprite backgroundPicture = null)

public void Create(string buttonTitle = "Button", Sprite backgroundPicture = null)

创建函数，参数buttonTitle为button.text.text的内容，backgroundPicture为button.image.sprite的图片

用途：动态更改按钮

### Create(UnityAction call, string buttonTitle = "Button")

public void Create(UnityAction call, string buttonTitle = "Button")

创建函数，参数call为button.ocClick绑定的委托【此委托是添加到委托列表中的，并非直接赋值】,buttonTitle为button.text.text的内容

用途：动态更改按钮

### Create(UnityAction call, string buttonTitle = "Button", Sprite backgroundPicture = null)

public void Create(UnityAction call, string buttonTitle = "Button", Sprite backgroundPicture = null)

创建函数，参数call为button.ocClick绑定的委托【此委托是添加到委托列表中的，并非直接赋值】，buttonTitle为button.text.text的内容，backgroundPicture为button.image.sprite的图片

用途：动态更改按钮

### OnPointerEnter(PointerEventData eventData)

public override void OnPointerEnter(PointerEventData eventData)

鼠标介入函数，参数 eventData为每次触摸事件都会创建其中一个包含所有相关信息的事件【此函数为虚函数】

用途：？？？

### OnPointerExit(PointerEventData eventData)

鼠标离开函数，参数 eventData为每次触摸事件都会创建其中一个包含所有相关信息的事件【此函数为虚函数】

用途：？？？

## IMCToggle

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### alpha

public override float alpha

透明度属性【取值范围0~1】【该属性为虚属性】

用途：更改组件的透明度

### interact

public override bool interact

交互开关属性【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【该属性为虚属性】

用途：判断该控件是否需要射线检测

### Initialize()

public override void Initialize()

重写初始化函数，使其只能进行一次初始化操作【该函数为虚函数】

用途：需要初始化的代码在重写该函数后添加

### label

public IMCText label

标签属性，获取该控件下的标签控件【该属性为只读属性】

用途：更改标签内容

### AddListener(UnityAction<bool> action)

public void AddListener(UnityAction<bool> action)

添加侦听委托函数，参数action为需要添加的侦听委托

用途：动态添加委托函数

### RemoveListener(UnityAction<bool> action)

public void RemoveListener(UnityAction<bool> action)

删除侦听委托函数，参数action为需要删除的侦听委托

用途：动态删除委托函数

### RemoveAllListener()

public void RemoveAllListener()

删除全部侦听委托函数

用途：动态删除全部委托函数

### isPlayAudio

public bool isPlayAudio

是否播放属性

用途：更改属性确定该控件是否有点击音效

### clip

public AudioClip clip

音频属性

用途：添加该音效则按钮触发当前音效否则触发IMCCanvas组件下的“统一”按钮音效

## IMCSlider

### value

public virtual float value

数值属性，【更改set方法，赋值时不会触发回调。更改为鼠标点击后触发回调】【该属性为虚属性】

用途：动态更改滑动条状态

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### alpha

public override float alpha

透明度属性【取值范围0~1】【该属性为虚属性】

用途：更改组件的透明度

### interact

public override bool interact

交互开关属性【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【该属性为虚属性】

用途：判断该控件是否需要射线检测

### Initialize()

public override void Initialize()

重写初始化函数，使其只能进行一次初始化操作【该函数为虚函数】

用途：需要初始化的代码在重写该函数后添加

## IMCScrollbar

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### alpha

public override float alpha

透明度属性【取值范围0~1】【该属性为虚属性】

用途：更改组件的透明度

### interact

public override bool interact

交互开关属性【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【该属性为虚属性】

用途：判断该控件是否需要射线检测

### Initialize()

public override void Initialize()

重写初始化函数，使其只能进行一次初始化操作【该函数为虚函数】

用途：需要初始化的代码在重写该函数后添加

## IMCDropdown

### OptionData

public class OptionData

选项数据类【该类为内部类】

用途：用于设置dropdown下每个Item的数据类

### OptionData.itemToggle

public IMCToggle itemToggle

Item toggle（item自身）控件属性

用途：设置item上的toggle控件

### OptionData.itemLabel

public IMCText itemLabel

Item Label控件属性

用途：设置item控件下ItemLabel 控件

### OptionData.itemBackground

public IMCImage itemBackground

Item Background控件属性

用途：设置item控件下Item Background控件

### OptionData.itemCheckmark

public IMCImage itemCheckmark

Item Checkmark控件属性

用途：设置item控件下Item Checkmark控件

### ShowItemAction

public UnityAction ShowItemAction = null;

Item显示委托，显示下拉列表时触发的委托回调

用途：显示下拉列表时需要触发事件绑定此委托

### HideItemAction

public UnityAction<int> HideItemAction = null;

Item隐藏委托，隐藏下拉列表时触发的委托回调

用途：隐藏下拉列表时需要触发事件绑定此委托

### onValueChanged

public DropdownEvent onValueChanged

值发生变化事件，当value发生变化时触发的事件回调【当value没有发生变化的时候不触发事件】

用途：value变换需要触发事件绑定此事件

### onHideChanged

public DropdownEvent onHideChanged

值发生变化事件，当value被赋值时触发的事件回调【无论手动触发还是代码触发】

用途：value被赋值时触发事件绑定此事件

### onSelect

public DropdownEvent onSelect

选择事件，当下拉列表显示时，修改value触发事件回调

用途：显示下拉列表时，value被赋值时需要触发事件绑定此事件

### isShow

public bool isShow

是否显示属性，是否显示下拉列表

用途：用于判断当前下拉列表是否显示

### alpha

public override float alpha

透明度属性【取值范围0~1】【该属性为虚属性】

用途：更改组件的透明度

### interact

public override bool interact

交互开关属性【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【该属性为虚属性】

用途：判断该控件是否需要射线检测

## IMCInputField

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### alpha

public override float alpha

透明度属性【取值范围0~1】【该属性为虚属性】

用途：更改组件的透明度

### interact

public override bool interact

交互开关属性【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【该属性为虚属性】

用途：判断该控件是否需要射线检测

### Initialize()

public override void Initialize()

重写初始化函数，使其只能进行一次初始化操作【该函数为虚函数】

用途：需要初始化的代码在重写该函数后添加

## IMCCanvas

### Name

public string Name = "";

名字变量

用途：可以把它看做是一个标签，可以通过这个名字查找对应的canvas

### buttonClickClip

public AudioClip buttonClickClip;

按钮点击音效

用途：可以使需要触发点击音效的按钮播放这个音效

### forms

public List<IMCForm> forms = new List<IMCForm>();

IMCForm集合

用途：可以通过该集合查找IMCForm

### PlayAudio(AudioClip clip)

public void PlayAudio(AudioClip clip)

播放音频函数【外界调用】

用途：播放按钮点击音效

### RegisterForm(IMCForm form)

public void RegisterForm(IMCForm form)

添加form函数，参数form为需要添加的对应form

用途：可以动态向forms集合中添加form

### UnRegisterForm(IMCForm form)

public void UnRegisterForm(IMCForm form)

移除form函数，参数form为需要移除的对应form

用途：可以动态从forms 集合中移除form

### FindFormByName(string name)

public IMCForm FindFormByName(string name)

通过form的name属性获取form函数，参数name为目标form的name属性【如果没有符合条件的元素，返回null】

用途：可以查找form

### FindFormsByName(string name)

public List<IMCForm> FindFormsByName(string name)

通过form的name属性获取form集合函数，参数name为目标form的name属性【如果没有符合条件的元素，返回长度为0的IMCForm类型的集合】

用途：可以查找多个同名form

### FindFormByByte<T>() where T:IMCForm

public T FindFormByByte<T>() where T:IMCForm

通过泛型查找forms集合中的form函数【限制泛型T类型为IMCForm的子类】

用途：可以查找form

### FindFormsByByte<T>() where T : IMCForm

public List<T> FindFormsByByte<T>() where T : IMCForm

通过form的类型获取form集合函数，泛型T为目标form的类型【如果没有符合条件的元素，返回长度为0的IMCForm类型的集合】

用途：可以查找多个同类型form

### RaycastSwitch(bool isSwitch)

public void RaycastSwitch(bool isSwitch)

射线开关函数，参数isSwitch为开关变量

用途：可以设置当前IMCCanvas下的所有ui能否被射线检测到

## IMCMessageBox

### MessageBoxButtonAttribute

public struct MessageBoxButtonAttribute

消息框按钮属性结构体

用途：？？？

### background

public IMCImage background;

背景图对象

用途：对该控件的背景图进行操作

### title

public IMCText title;

标题对象

用途：对该控件的标题进行操作

### content

public IMCText content;

内容对象

用途：对该控件的文字内容进行操作

### buttonParent

public GameObject buttonParent;

按钮父级（载体）对象

用途：对三个按钮统一进行管理

### closeButton

public IMCButton closeButton;

关闭按钮对象

用途：可以获取该对象对其进行绑定回调，隐藏显示等操作

### closeObject

被关闭对象

用途：关闭MessageBox时可以销毁（Destroy）一个目标对象

### btns

public List<IMCButton> btns = new List<IMCButton>();

buttonParent 下的Button集合【在Awake时获取】

用途：可以对集合内元素进行自定义操作

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### alpha

public override float alpha

透明度属性【取值范围0~1】【影响子物体】【该属性为虚属性】

用途：更改自身的透明度

### interact

public override bool interact

交互开关属性【影响子物体】【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【影响子物体】【该属性为虚属性】

用途：判断该控件是否需要射线检测

### Deploy(UnityAction<int> choiceBack, string messageBoxTitle, string messageBoxContent, Sprite messageBoxBackground, MessageBoxButtonAttribute[] messageBoxAttribute)

public void Deploy(UnityAction<int> choiceBack, string messageBoxTitle, string messageBoxContent, Sprite messageBoxBackground, MessageBoxButtonAttribute[] messageBoxAttribute)

部署MessageBox函数，参数choiceBack为有参回调，messageBoxTitle为控件标题，messageBoxContent为控件文字内容，messageBoxBackground为控件背景图，messageBoxAttribute为对每个btns数组里的按钮单独属性赋值

用途：设置MessageBox

### Deploy(UnityAction<int> choiceBack)

public void Deploy(UnityAction<int> choiceBack)

部署MessageBox函数，参数choiceBack为有参回调【返回的参数是btns数组中对应的button索引号】

用途：设置MessageBox

### Deploy(UnityAction<int> choiceBack, string messageBoxTitle, string messageBoxContent)

public void Deploy(UnityAction<int> choiceBack, string messageBoxTitle, string messageBoxContent)

部署MessageBox函数，参数choiceBack为有参回调【返回的参数是btns数组中对应的button索引号】，messageBoxTitle为控件标题，messageBoxContent为控件文字内容

用途：设置MessageBox

### Deploy(UnityAction<int> choiceBack, string messageBoxTitle, string messageBoxContent, Sprite messageBoxBackground)

public void Deploy(UnityAction<int> choiceBack, string messageBoxTitle, string messageBoxContent, Sprite messageBoxBackground)

部署MessageBox函数，参数choiceBack为有参回调【返回的参数是btns数组中对应的button索引号】，messageBoxTitle为控件标题，messageBoxContent为控件文字内容，messageBoxBackground为控件背景图

用途：部署MessageBox

### SetButton(int buttonIndex, string buttonTitle = "", Sprite buttonSprite = null, UnityAction buttonAction = null)

public void SetButton(int buttonIndex, string buttonTitle = "", Sprite buttonSprite = null, UnityAction buttonAction = null)

设置按钮函数，参数buttonIndex为按钮所在btns数组中的索引查找对应的按钮，buttonTitle 为按钮的text控件内容，buttonSprite 为按钮的image控件的sprite图片，buttonAction 为按钮所需要添加绑定的委托【针对于btns数组中的函数，不包含MessageBox的closeBtn控件】

用途：设置btns数组中的按钮

### SetButton(int buttonIndex, MessageBoxButtonAttribute buttonAttribute)

public void SetButton(int buttonIndex, MessageBoxButtonAttribute buttonAttribute)

设置按钮函数，参数buttonIndex为按钮所在btns数组中的索引查找对应的按钮，参数buttonAttribute为消息框按钮属性结构体【针对于btns数组中的函数，不包含MessageBox的closeBtn控件】

用途：设置btns数组中的按钮

### SetButton(IMCButton buttonObject, string buttonTitle = "", Sprite buttonSprite = null, UnityAction buttonAction = null)

public void SetButton(IMCButton buttonObject, string buttonTitle = "", Sprite buttonSprite = null, UnityAction buttonAction = null)

设置按钮函数，参数buttonObject为目标按钮，buttonTitle 为按钮的text控件内容，buttonSprite 为按钮的image控件的sprite图片，buttonAction 为按钮所需要添加绑定的委托【针对于btns数组中的函数，不包含MessageBox的closeBtn控件】

### RemovButtonElement(int buttonIndex)

public void RemovButtonElement(int buttonIndex)

移除按钮元素函数，参数buttonIndex为btns数组中的索引

用途：销毁btns中的指定元素

### RemoveButtonElement(IMCButton buttonObject)

public void RemoveButtonElement(IMCButton buttonObject)

移除按钮元素函数，参数buttonObject为btns数组中的按钮对象

用途：销毁btns中的置指定元素

## IMCCascadeDropdown

public class IMCCascadeDropdown : IMCUIBehaviour

级联·下拉控件

### dropDowns

public List<IMCDropdown> dropDowns

下拉控件集合

用途:可以自定义集合内的元素

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### alpha

public override float alpha

透明度属性【取值范围0~1】【影响dropDowns集合中的元素】【该属性为虚属性】

用途：更改自身的透明度

### interact

public override bool interact

交互开关属性【影响dropDowns集合中的元素】【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【影响dropDowns集合中的元素】【该属性为虚属性】

用途：判断该控件是否需要射线检测

### GetDropDownsIndex(IMCDropdown dropDown)

public int GetDropDownsIndex(IMCDropdown dropDown)

获取下拉控件索引函数，参数dropdown为dropDowns集合中的对应下拉控件【如果没有对应索引，则返回-1】

用途：通过参数dropDown返回对应dropDowns数组中的索引

### currentDropdown

public IMCDropdown currentDropdown

当前下拉控件属性

用途：获取当前正在工作的下拉控件

## IMCSwitchButton

public class IMCSwitchButton : IMCSlider

【switchButton为IOS的官方命名】

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### alpha

public override float alpha

透明度属性【取值范围0~1】【影响子物体】【该属性为虚属性】

用途：更改自身的透明度

### interact

public override bool interact

交互开关属性【影响子物体】【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【影响子物体】【该属性为虚属性】

用途：判断该控件是否需要射线检测

### content

public IMCText content

内容控件属性

用途：自定义该控件显示内容

### onClick

public switchButtonEvent onClick = new switchButtonEvent();

点击委托【该委托为有参委托，回调参数为switchButton当前的isOn】

用途：用于动态绑定委托

### moveSpeed

public float moveSpeed

移动速度属性

用途：用于动态设置Handle的移动速度

### isOn

public bool isOn

开关属性

用途：用于动态设置switchButton控件的开关

### onContent

public string onContent

打开时显示文字属性

用途：用于设置switchButton控件isOn为true时显示的文字内容

### offContent

public string offContent

关闭时显示文字属性

用途：用于设置switchButton控件isOn为flase时显示的文字内容

### isContent

public bool isContent

是否打开文字显示属性

用途：用于设置switchButton控件的文字显示功能是否打开

### isColor

public bool isColor

是否打开颜色显示属性【默认颜色为白色】

用途：用于设置switchButton控件的颜色显示功能是否打开

### onColor

public Color onColor

打开时显示的颜色

用途：用于设置switchButton控件isOn为true时显示的颜色

### offColor

public Color offColor

关闭时显示的颜色

用途：用于设置switchButton控件isOn为flase时显示的颜色

### canvasGroup

public CanvasGroup canvasGroup

画布组控件属性【该属性为只读属性】

用途：设置画布组控件自定义效果

### IsOnToggle(bool ison)

public void IsOnToggle(bool ison)

isOn开关函数,参数isOn为开关函数【因为继承slider的原因，如果需要代码调用开关的话，要用此函数】

用途：动态设置isOn的值

## IMCOrderShowButtons

public class IMCOrderShowButtons : IMCUIBehaviour

【翻页按钮组】

### targets

public List<IMCUIBehaviour> targets;

目标集合

用途：用于动态操作目标集合

### leftBtn

public IMCButton leftBtn;

左按钮

用途：向左翻页

### rightBtn

public IMCButton rightBtn;

右按钮

用途：向右翻页

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### alpha

public override float alpha

透明度属性【取值范围0~1】【影响子物体】【该属性为虚属性】

用途：更改自身的透明度

### interact

public override bool interact

交互开关属性【影响子物体】【该属性为虚属性】

用途：？？

### raycast

public override bool raycast

射线开关属性【影响子物体】【该属性为虚属性】

用途：判断该控件是否需要射线检测

### LeftBtnClick()

public void LeftBtnClick()

左按钮事件函数

用途：向左翻页

### RightBtnClick()

public void RightBtnClick()

右按钮事件函数

用途：向右翻页

### ResetTargets(int index = 0)

public void ResetTargets(int index = 0)

刷新目标函数，参数index为刷新后显示的target目标索引

用途：用于重置显示状态

## IMCHyperlinksLabel

public class IMCHyperlinksLabel :IMCText ,IPointerDownHandler,IPointerClickHandler

### onHrefClick

public HrefClickEvent onHrefClick

超链接点击事件【默认绑定打开OpenURL函数】

用途：用于绑定超链接被点击时触发的委托

### hyperlinksLaberFontSize

public int hyperlinksLaberFontSize

超链接文字大小属性

用途：更改超链接字体大小

### hyperlinksLaberColor

public Color hyperlinksLaberColor

超链接字体颜色属性

用途：更改超链接文字颜色

### SetHyperlinksContent(string url, string content)

public void SetHyperlinksContent(string url, string content)

设置超链接内容函数，参数url为url地址，content为代替url显示的内容

用途：用于设置超链接控件内容

### SetHyperlinksContent(string url, string content, Color contentColor, int contentFontSize)

public void SetHyperlinksContent(string url, string content, Color contentColor, int contentFontSize)

设置超链接内容函数，参数url为url地址，content为代替url显示的内容，contentColor为content字体颜色，contentFontSize为content字体大小

用途：用于设置超链接控件内容

### SetHyperlinksContent(string url, string content, string contentColor, int contentFontSize)

public void SetHyperlinksContent(string url, string content, string contentColor, int contentFontSize)

设置超链接内容函数，参数url为url地址，content为代替url显示的内容，contentColor为content字体颜色，contentFontSize为content字体大小

用途：用于设置超链接控件内容

## IMCProgressBar

public class IMCProgressBar : IMCUIBehaviour

### Awake()

protected override void Awake()

重写Awake函数，更改自身控件类型枚举和容器类型枚举变量【该函数为虚函数】

用途：重写该函数设置自定义功能

### animationStyle

public AnimationSyle animationStyle = AnimationSyle.None;

动画类型枚举

用途：更改动画的风格

### value

public float value

数值属性

用途：对于个别动画状态更改数值可以显示进度条百分比

### sprite

public Sprite sprite

精灵图属性【此属性为只读属性】

用途：可以更改进度条图片

### background

public Sprite background

背景图属性【此属性为只读属性】

用途：可以更改背景图

### tempView

public IMCImage tempView

临时视图image控件属性，是sprite属性的载体【此属性为只读属性】

用途：更改进度条等一系列自定义操作

### backgroundImage

public IMCImage backgroundImage

背景图image控件属性【此属性为只读属性】

用途：更改背景

### rotateSpeed

public float rotateSpeed = -100f;

旋转速度变量，旋转动画状态的旋转速度

用途：更改旋转速度，以及旋转方向

### isTempViewCustomSize

public bool isTempViewCustomSize = false;

TempView自定义大小开关变量

用途：此布尔变量为true时，tempView控件size会设置为tempViewSize

### tempViewSize

public Vector2 tempViewSize

TempView大小属性【该属性在isTempViewCustomSize 为true时才被触发】

用途：更改TempView控件大小

### isTempViewPosition

public bool isTempViewPosition = false;

TempView自定义位置开关变量

用途：此布尔变量为true时，tempView控件position会设置为tempViewPosition

### tempViewPosition

public Vector3 tempViewPosition

TempView位置属性【该属性在isTempViewPosition为true时才被触发】

用途：更改TempView控件位置

### Begin(Sprite tempViewSprite = null, Sprite backgroundSprite = null)

public void Begin(Sprite tempViewSprite = null, Sprite backgroundSprite = null)

开始函数，参数tempViewSprite为进度条图片，backgroundSprite为背景图片

用途：启动进度条控件动画

### Begin(AnimationStyle style, Sprite tempViewSprite = null, Sprite backgroundSprite = null)

public void Begin(AnimationStyle style, Sprite tempViewSprite = null, Sprite backgroundSprite = null)

开始函数，参数style为动画类型枚举，参数tempViewSprite为进度条图片，backgroundSprite为背景图片

用途:启动进度条控件动画

### Stop(bool isDestroy=false)

public void Stop(bool isDestroy=false)

关闭进度条，参数isDestroy为是否销毁自身组件

用途：关闭进度条动画控件显示

## IMCUIContainer

public class IMCUIContainer : IMCUIBehaviour

【容器类】

### canvasGroup

public CanvasGroup canvasGroup

画布组控件属性【此属性为只读属性】

用途：更改控件及子物体射线开关，交互开关以及alpha数值

### image

public IMCImage image

图像控件属性【此属性为只读属性】

用途：自定义自身图片的显示状态

### raycast

public override bool raycast

射线开关【影响子集】

用途：用于设置该控件下的所有子物体的射线检测是否响应

### alpha

public override float alpha

透明度开关【影响子集】

用途：用于设置该控件下所有子物体的透明度

### interact

public override bool interact

交互开关【影响子集】

用途：用于设置该控件下所有子物体的交互操作是否响应

### Initialize()

public override void Initialize()

初始化函数【此函数为虚函数】

用途：重写此函数个性化初始化操作

### GetContainerObject()

public override IMCUIContainer GetContainerObject()

获得容器函数【此函数为虚函数】

用途：获得自身

### AddControl(IMCUIBehaviour control)

public virtual void AddControl(IMCUIBehaviour control)

添加子物体函数，参数control为需要添加的子物体【此函数为虚函数】

用途： 向容器添加子物体

### AddControl(IMCUIBehaviour control, Vector3 pos)

public virtual void AddControl(IMCUIBehaviour control, Vector3 pos)

添加子物体函数，参数control为需要添加的子物体，pos为相对于父级的anchoredPosition3D位置【此函数为虚函数】

用途：向容器添加子物体

### AddControls(List<IMCUIBehaviour> list)

public virtual void AddControls(List<IMCUIBehaviour> list)

添加子物体集合函数，参数list为需要添加的子物体集合【此函数为虚函数】

用途：向容器添加子物体

### AddControls(List<IMCUIBehaviour> list, List<Vector2> posList)

public virtual void AddControls(List<IMCUIBehaviour> list, List<Vector2> posList)

添加子物体集合函数，参数list为需要添加的子物体集合，posList为list元素对应的相对于父级的anchoredPosition3D位置【此函数为虚函数】

用途：向容器添加子物体

### RemoveControl(IMCUIBehaviour control)

移除子物体函数，参数control为需要被移除的子物体【此函数为虚函数】

用途：移除controls数组中对应的元素

### RemoveControlAndDestroy(IMCUIBehaviour control)

移除并销毁子物体函数，参数control为需要被移除并销毁的子物体【此函数为虚函数】

用途：移除并销毁control数组中对应的元素

### RemoveAllControl()

public virtual void RemoveAllControl()

移除全部子物体函数【此函数为虚函数】

用途：移除controls数组中全部元素

### RemoveAllControlAndDestroy()

public virtual void RemoveAllControlAndDestroy()

移除并销毁全部子物体函数【此函数为虚函数】

用途：移除并销毁contorls数组中的全部元素

### FindControl<T>()

public virtual T FindControl<T>()where T:IMCUIBehaviour

查找控件函数，泛型T为查找的控件类型【通过正向查找，返回第一个符合条件的元素】【此函数为虚函数】

用途：查找子物体

### FindControlByCustomID(string customID)

public virtual IMCUIBehaviour FindControlByCustomID(string customID)

通过控件CustomID查找函数，参数customID为控件控件的customID【通过正向查找，返回第一个符合条件的元素】【此函数为虚函数】

用途：查找子物体

### FindControlByCustomID<T>(string customID)

public virtual T FindControlByCustomID<T>(string customID) where T : IMCUIBehaviour

通过控件CustomID查找函数，泛型T为查找的控件类型，customID为控件控件的customID【通过正向查找，返回第一个符合条件的元素】【此函数为虚函数】

用途：查找子物体

### FindControlByName(string name)

public virtual IMCUIBehaviour FindControlByName(string name)

通过控件name查找函数，参数name为控件的name变量值【gameObject.name】【通过正向查找，返回第一个符合条件的元素】【此函数为虚函数】

用途：查找子物体

### FindControlByName<T>(string name)

public virtual T FindControlByName<T>(string name) where T : IMCUIBehaviour

通过控件name查找函数，泛型T为查找的控件类型，参数name为控件的name变量值【gameObject.name】【通过正向查找，返回第一个符合条件的元素】【此函数为虚函数】

用途：查找子物体

### FindControlByNameAndCustomID(string name, string customID)

public virtual IMCUIBehaviour FindControlByNameAndCustomID(string name, string customID)

通过控件name与customID查找函数，参数name为控件的name变量值【gameObject.name】，customID为控件控件的customID【通过正向查找，返回第一个符合条件的元素】【此函数为虚函数】

用途：查找子物体

### FindControlByNameAndCustomID<T>(string name, string customID)

public virtual T FindControlByNameAndCustomID<T>(string name, string customID) where T : IMCUIBehaviour

通过控件name与customID查找函数，泛型T为控件的类型，参数name为控件的name变量值【gameObject.name】，customID为控件控件的customID【通过正向查找，返回第一个符合条件的元素】【此函数为虚函数】

用途：查找子物体

### FindControls<T>()

public virtual List<T> FindControls<T>() where T : IMCUIBehaviour

查找子物体集合函数，泛型T为控件类型【此函数为虚函数】

用途：查找子物体集合

### FindControlsByCustomID(string customID)

public virtual List<IMCUIBehaviour> FindControlsByCustomID(string customID)

通过控件customID查找函数，参数customID为控件的customID【此函数为虚函数】

用途：查找子物体集合

### FindControlsByCustomID<T>(string customID)

public virtual List<T> FindControlsByCustomID<T>(string customID) where T : IMCUIBehaviour

通过控件customID查找函数，泛型T为控件类型，参数customID为控件的customID【此函数为虚函数】

用途：查找子物体集合

### FindControlsByName(string name)

public virtual List<IMCUIBehaviour> FindControlsByName(string name)

通过控件name查找函数，参数name为控件的name值【gameObject.name】【此函数为虚函数】

用途：查找子物体集合

### FindControlsByName<T>(string name)

public virtual List<T> FindControlsByName<T>(string name) where T : IMCUIBehaviour

通过控件name查找函数，泛型T为控件类型，参数name为控件的name值【gameObject.name】【此函数为虚函数】

用途：查找子物体集合

### FindControlsByNameAndCustomID(string name, string customID)

public virtual List<IMCUIBehaviour> FindControlsByNameAndCustomID(string name, string customID)

通过控件name与custiomID查找函数，参数name为控件的name值【gameObject.name】，customID为控件的customID【此函数为虚函数】

用途：查找子物体集合

### FindControlsByNameAndCustomID<T>(string name, string customID)

public virtual List<T> FindControlsByNameAndCustomID<T>(string name, string customID) where T : IMCUIBehaviour

通过控件name与custiomID查找函数，泛型T为控件类型，参数name为控件的name值【gameObject.name】，customID为控件的customID【此函数为虚函数】

用途：查找子物体集合

### FindControlByIntanceID(int intanceID)

public virtual IMCUIBehaviour FindControlByIntanceID(int intanceID)

通过控件InstanceID查找函数，参数instanceID为控件intanceID【此函数为虚函数】

用途：查找子物体

### InitializeChildControls()

protected virtual void InitializeChildControls()

获取子物体函数【此函数为虚函数】

用途：？？？

### GetChildControls()

public List<IMCUIBehaviour> GetChildControls()

获取子物体集合函数

用途：获取子物体集合

### CanvasGroupSwitch(bool m\_switch)

protected void CanvasGroupSwitch(bool m\_switch)

画布组开关函数，参数m\_switch为开关变量

用途：设置当前组件画布组开关，【包括射线，交互，透明度】

### MoveHouse(IMCUIBehaviour control, IMCUIContainer form, IMCUIContainer to, Vector2 pos, int tabViewIndex = 0)

public static void MoveHouse(IMCUIBehaviour control, IMCUIContainer form, IMCUIContainer to, Vector2 pos, int tabViewIndex = 0)

搬家函数，将control对象从form容器对象中移至to容器对象中【如果control不存在于form容器controls数组中，程序不会执行任何操作。如果to容器是tabView类型容器，还需要指定tabViewIndex传入具体的tabcontent索引。】【此函数为静态函数】

用途：子物体批量更换父级时调用此函数

## IMCForm

public class IMCForm : IMCUIContainer

### canvas

public IMCCanvas canvas;

Canvas变量

用途：获取当前form的IMCCanvas

### showOnAwake

public bool showOnAwake

开始立刻显示属性

用途：此属性决定程序运行后是否自动显示当前form

### showTargetPosition

public Vector3 showTargetPosition

Show状态目标位置属性

用途：show状态时form的显示的anchoredPosition3D位置

### closeTargetPosition

public Vector3 closeTargetPosition

Close状态目标位置属性

用途：Close状态时form的显示的anchoredPosition3D位置

### isBlockerToggle

public bool isBlockerToggle

是否需要阻挡器属性

用途：show状态时是否需要阻挡器

### isDontDestroy

public bool isDontDestroy = false;

是否跳转场景销毁变量

用途：为true时，跳转场景后form不销毁

### blockerShowStyle

public IMCBlocker.ShowStyleEnum blockerShowStyle = IMCBlocker.ShowStyleEnum.Transparent;

阻拦器显示风格枚举变量【当isBlockerToggle为true时才会有用】

用途：设置阻拦器显示风格

### InitializeAction

public UnityAction InitializeAction;

初始化委托

用途：初始化时需要执行代码绑定此委托

### ShowAction

public UnityAction ShowAction;

显示委托

用途：show状态时需要执行代码绑定此委托

### CloseAction

public UnityAction CloseAction;

关闭委托

用途：close状态时需要执行代码绑定此委托

### UnInitAction

public UnityAction UnInitAction;

注销委托

用途：注销状态时需要执行代码绑定此委托

### blockerCallBack

public UnityAction blockerCallBack;

阻拦器回调委托

用途：阻拦器触发后需要执行代码绑定此委托

### showCallBack

public UnityEvent showCallBack;

显示事件

用途：show状态时需要执行代码绑定此事件

### closeCallBack

public UnityEvent closeCallBack;

关闭事件

用途：close状态时需要执行代码绑定此事件

### Show()

public virtual void Show()

显示函数【此函数为虚函数】

用途：需要显示控件时调用此函数

### Show(Vector3 showTargetPosition, bool blockerToggle = false, IMCBlocker.ShowStyleEnum showStyle = IMCBlocker.ShowStyleEnum.Transparent, UnityEvent showCallBack = null, UnityAction blockerCallBack = null)

public virtual void Show(Vector3 showTargetPosition, bool blockerToggle = false, IMCBlocker.ShowStyleEnum showStyle = IMCBlocker.ShowStyleEnum.Transparent, UnityEvent showCallBack = null, UnityAction blockerCallBack = null)

显示函数，参数showTargetPosition为show状态时控件的anchoredPosition3D位置，blockerToggle 为是否需要阻拦器，showStyle 为阻拦器显示状态时的枚举【blockerToggle 为flase时不生效】，showCallBack 为显示控件之后的回调委托，blockerCallBack 为阻拦器被触发后的回调委托【blockerToggle 为flase时不生效】【此函数为虚函数】

用途：需要显示控件时调用此函数

### Close()

public virtual void Close()

关闭函数【此函数为虚函数】

用途：需要关闭控件时调用此函数

### Close(Vector3 closeTargetPosition, UnityEvent closeCallBack = null)

public virtual void Close(Vector3 closeTargetPosition, UnityEvent closeCallBack = null)

关闭函数，参数closeTargetPosition为close状态时控件的anchoredPosition3D位置，closeCallBack为阻拦器被触发后的回调委托【此函数为虚函数】

用途：需要关闭控件时调用此函数

### UnInit(int time = 0)

public override void UnInit(int time = 0)

注销函数，参数time为延迟销毁时间【此函数为虚函数】

用途：注销/销毁控件时调用此函数

### Awake()

protected override void Awake()

Awake函数，重写此函数，更改控件类型枚举【此函数为虚函数】

用途：在Awake时需要执行代码重写此函数

### Initialize()

public override void Initialize()

初始化函数，调用controls数组中元素的初始化函数【此函数为虚函数】

用途：在初始化时需要执行代码重写此函数

### OnDestroy()

protected override void OnDestroy()

在销毁之前函数【此函数为虚函数】

用途：在销毁之前需要执行代码重写此函数

## IMCScrollRect

public class IMCScrollRect : IMCUIContainer, IInitializePotentialDragHandler, IBeginDragHandler, IEndDragHandler, IDragHandler, IScrollHandler, ICanvasElement, ILayoutElement, ILayoutGroup, IPointerDownHandler

### parentScrollRect

public IMCScrollRect parentScrollRect

父级ScrollRect属性

用途：多层级ScrollRect嵌套赋值此属性

### Awake()

protected override void Awake()

Awake函数，更改控件类型枚举【此函数为虚函数】

用途：在Awake时需要执行代码重写此函数

### InitializeChildControls()

protected override void InitializeChildControls()

初始化获取controls元素函数【此函数为虚函数】

用途：？？？

### Initialize()

初始化函数，调用controls数组中元素的初始化函数【此函数为虚函数】

用途：在初始化时需要执行代码重写此函数

## IMCTabView

public class IMCTabView : IMCUIContainer

### toggles

public List<IMCTabViewToggle> toggles = new List<IMCTabViewToggle>();

Toggle集合

用途：增删toggles集合元素，查找指定元素

### contents

public List<IMCUIBehaviour> contents = new List<IMCUIBehaviour>();

contents 集合

用途：增删contents 集合元素，查找指定元素

### Awake()

protected override void Awake()

Awake函数，更改控件类型枚举【此函数为虚函数】

用途：在Awake时需要执行代码重写此函数

### OnEnable()

protected override void OnEnable()

在激活之前函数，对toggles 集合和contents 进行初始化操作【此函数为虚函数】

用途：在控件激活之前需要执行代码重写此函数

### TabViewControlShowOrClose(bool nuclearBombSwitch, IMCTabViewToggle tvt)

public void TabViewControlShowOrClose(bool nuclearBombSwitch, IMCTabViewToggle tvt)

选项卡视图控制显示或关闭函数，参数nucleraBombSwitch为每个toggle的ison值，tvt为指定的IMCTabViewToggle类型控件。此函数是自定义广播函数【此函数为内部调用函数】

用途：？？？

### AddControl(int index, IMCUIBehaviour control)

public void AddControl(int index, IMCUIBehaviour control)

添加control元素函数，参数index为contents 集合索引，controls为需要添加的控件

用途：动态添加controls控件调用此函数

### AddControl(int index, IMCUIBehaviour control, Vector3 pos)

### public void AddControl(int index, IMCUIBehaviour control, Vector3 pos)

添加control元素函数，参数index为contents 集合索引，controls为需要添加的控件，pos为controls的anchoredPosition3D位置

用途：动态添加controls控件调用此函数

### AddControl(IMCUIBehaviour content, IMCUIBehaviour control)

public void AddControl(IMCUIBehaviour content, IMCUIBehaviour control)

添加control元素函数，参数content为contents 集合中的元素，controls为需要添加的控件

用途：动态添加controls控件调用此函数

### AddControl(IMCUIBehaviour content, IMCUIBehaviour control, Vector2 pos)

public void AddControl(IMCUIBehaviour content, IMCUIBehaviour control, Vector2 pos)

添加control元素函数，参数content为contents 集合中的元素，controls为需要添加的控件，pos为controls的anchoredPosition3D位置

用途：动态添加controls控件调用此函数

### InitializeChildControls()

protected override void InitializeChildControls()

初始化获取controls元素函数【此函数为虚函数】

用途：？？？

### GetCurrentOperationContentGameObjectIndex()

public int GetCurrentOperationContentGameObjectIndex()

获取当前正在执行的content控件的索引函数

用途：获取当前显示的content控件索引

### SetShowContent(int index)

public void SetShowContent(int index)

设置显示content函数，参数index为需要显示的contents集合索引

用途：动态设置content的显示

## IMCGroup

public class IMCGroup : IMCUIContainer

【一个特殊的容器类型控件，因为Form不能嵌套Form，所以才会有Group容器控件，继承容器类型所有特性】

### Awake()

protected override void Awake()

Awake函数，更改控件类型枚举【此函数为虚函数】

用途：在Awake时需要执行代码重写此函数

## IMCBlocker

public class IMCBlocker : IMCUIBehaviour, IPointerClickHandler

### Instance

public static IMCBlocker Instance

单例属性【此属性为只读属性】

用途：调用非静态函数、属性

### Is\_blockerPointerClick

public bool Is\_blockerPointerClick

阻拦器是否完成点击属性【此属性为只读属性】

用途：判断阻拦器是否执行点击

### ShowStyleEnum

public enum ShowStyleEnum

显示风格枚举

用途：设置阻拦器的显示风格

### Create(List<IMCUIBehaviour> targetUis, ShowStyleEnum showStyle, UnityAction callback = null, bool isDestroy = true, Transform parent = null)

public IMCBlocker Create(List<IMCUIBehaviour> targetUis, ShowStyleEnum showStyle, UnityAction callback = null, bool isDestroy = true, Transform parent = null)

创建函数，参数targetUis为需要被阻拦器呈现的ui集合，showStyle为阻拦器的显示风格，callback为触发阻拦器的回调委托，isDestroy为触发阻拦器后是否销毁函数，parent为生成在指定控件下成为子物体

用途：创建阻拦器控件

### Awake()

protected override void Awake()

Awake函数，更改控件类型枚举，更改容器类型枚举【此函数为虚函数】

用途：在Awake时需要执行代码重写此函数

### OnDestroy()

protected override void OnDestroy()

在销毁之前函数【此函数为虚函数】

用途：在销毁之前需要执行代码重写此函数

### RemoveStack(IMCUIBehaviour removeTarget)

public void RemoveStack(IMCUIBehaviour removeTarget)

移除栈元素函数，参数removeTarget为需要移除的指定控件

用途：动态移除指定元素

### Screenshot()

public Sprite Screenshot()

截屏函数

用途：截屏

## IMCDebug

public class IMCDebug : IMCForm

【呼出手势：同一位置双击屏幕，随后点击屏幕从右往左滑动一段距离，然后再某一位置连续点击五下 AA←AAAAA】【此控件为resource加载，使用前确保resource文件夹中有此控件预制体】

### Instance

public static IMCDebug Instance

单例属性【此属性为只读属性】

用途：调用非静态函数、属性

### debugDatas

public List<IMCDebugCreateData> debugDatas;

debugDatas集合【0 debug 1 error 2 warning】

用途：？？？

### Log(string content, GameObject go = null)

public static void Log(string content, GameObject go = null)

打印日志函数，参数content为字符内容，go为目标组件【此函数为静态函数】

用途：打印日志

### LogError(string content, GameObject go = null)

public static void LogError(string content, GameObject go = null)

打印错误日志函数，参数content为字符内容，go为目标组件【此函数为静态函数】

用途：打印错误日志

### LogWarning(string content, GameObject go = null)

public static void LogWarning(string content, GameObject go = null)

打印警告日志函数，参数content为字符内容，go为目标组件【此函数为静态函数】

用途：打印警告日志

### Show()

public override void Show()

显示函数【此函数为虚函数】

用途：显示控件

### Close()

public override void Close()

关闭函数【此函数为虚函数】

用途：关闭控件

### Clear()

public void Clear()

清除函数，清除当前显示的content控件内容【此函数为按钮绑定函数】

用途：清除IMCDebug控制台内容

### AddFontSize()

public void AddFontSize()

放大字体函数，放大当前显示的content空间字体【此函数为按钮绑定函数】

用途：放大IMCDebug控制台字体内容

### ReduceFontSize()

public void ReduceFontSize()

缩小字体函数，放大当前显示的content空间字体【此函数为按钮绑定函数】

用途：缩小IMCDebug控制台字体内容

### tabView

public IMCTabView tabView;

tabView控件，【按钮操作台容器】

用途：？？？

## IMCDragMove

public class IMCDragMove : IMCUIBehaviour, IPointerDownHandler, IDragHandler

### panelRectTransform

public RectTransform panelRectTransform;

面板对象，被移动物体

用途：设置被拖拽物体

### rangeLimit

public bool rangeLimit = true;

范围限制变量

用途：设置范围限制

## IMCUIAnimation

public class IMCUIAnimation : MonoBehaviour

### MoveTo(IMCUIBehaviour obj, Vector3 targetPos, float time, UnityAction callback = null)

public IEnumerator MoveTo(IMCUIBehaviour obj, Vector3 targetPos, float time, UnityAction callback = null)

位移协程，参数obj为需要位移的控件，targetPos为需要位移的anchoredPosition3D位置，time为移动时间（秒），callback为移动到targetPos位置时回调委托

用途：位移

### RotationTo(IMCUIBehaviour obj, Vector3 targetAngle, float time, UnityAction callbacke = null)

public IEnumerator RotationTo(IMCUIBehaviour obj, Vector3 targetAngle, float time, UnityAction callbacke = null)

旋转协程，参数obj为需要旋转的控件，targetAngle为需要旋转的角度，time为旋转时间(秒），callback为旋转到targetAngle角度时回调委托

用途：旋转

### ScaleTo(IMCUIBehaviour obj, Vector3 targetScale, float time, UnityAction callback = null)

public IEnumerator ScaleTo(IMCUIBehaviour obj, Vector3 targetScale, float time, UnityAction callback = null)

缩放协程，参数obj为需要缩放的控件，targetScale为需要缩放的大小，time为缩放时间(秒），callback为缩放到targetScale大小时回调委托

用途：缩放

## Alp**haTo(IMCUIBehaviour obj, float alpha,float time,UnityAction callback=null)**

public IEnumerator AlphaTo(IMCUIBehaviour obj, float alpha,float time,UnityAction callback=null)

透明度协程【透明度渐变】，参数obj为需要缩放的控件，alpha为目标透明度值，time为缩放时间(秒），callback为渐变到alpha时回调委托

用途：透明度渐变

## IMCUIResouceManager

public class IMCUIResouceManager : MonoBehaviour

### LoadControl<T>(string path, Transform parent = null)

public static T LoadControl<T>(string path, Transform parent = null)where T:IMCUIBehaviour

加载控件函数，泛型T为IMCUIBehaviour限制类型控件，参数path为加载路径，parent为设置指定父级【加载控件的gameObject.name要和泛型T的类型名相同】【该函数为静态函数】

用途：加载控件

### LoadControl(string path, string name, Transform parent = null)

public static GameObject LoadControl(string path, string name, Transform parent = null)

加载控件函数，参数path为加载路径，name为具体的控件name【gameObject.name】，parent为设置指定父级【该函数为静态函数】

用途：加载控件

### LoadControl(string pathAndName, Transform parent = null)

public static GameObject LoadControl(string pathAndName, Transform parent = null)

加载控件函数，参数pathAndName为加载路径和控件name【gameObject.name】，parent为设置指定父级【该函数为静态函数】

用途：加载控件