EVIS 3D SDK_URP 使用说明 V1.03

2023/05/22

上海易维视科技有限公司

免责声明

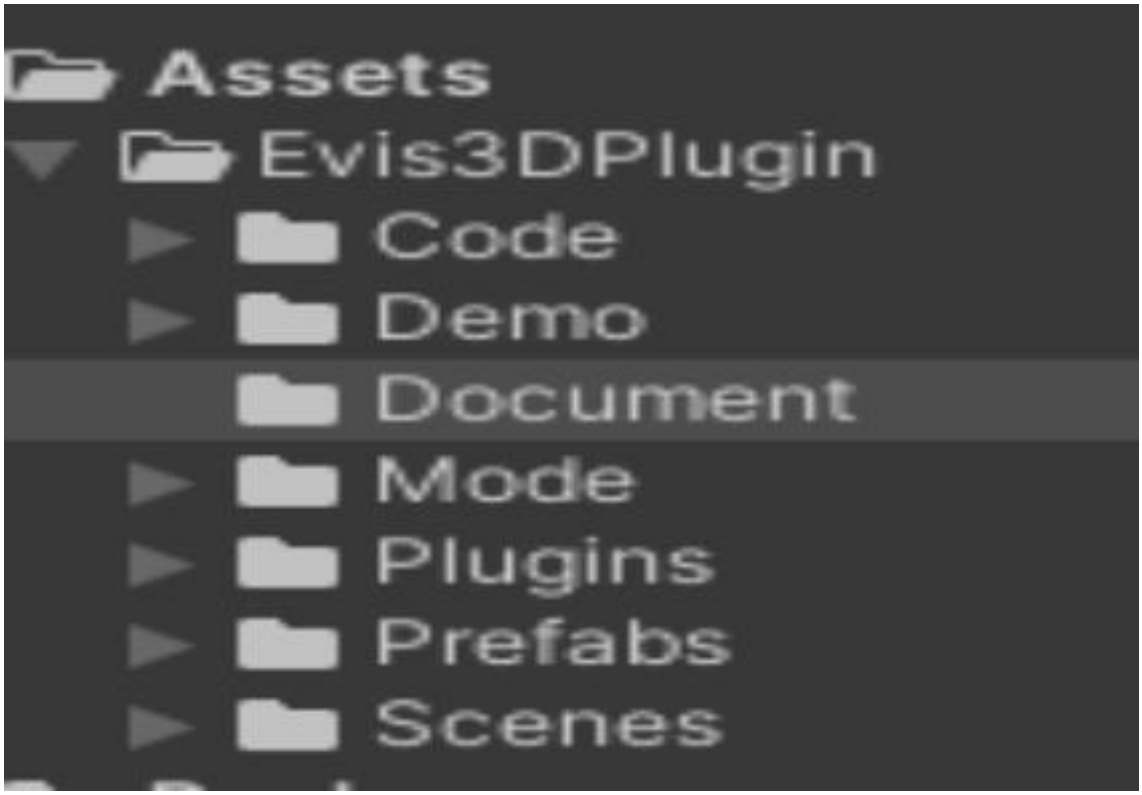
本手册依据现有信息制作，其内容如有更改，恕不另行通知。对上海易维视科技有限公司产品和服务的唯一担保在随产品和服务一起提供的明示保修声明中列出。此处的任何信息不应解释为构成了附加担保。上海易维视科技有限公司在编写该手册的时候已尽最大努力保证其内容准确可靠，不对本手册中的技术或者编辑遗漏、不准确、或错误导致的损失和损害承担责任。

目录

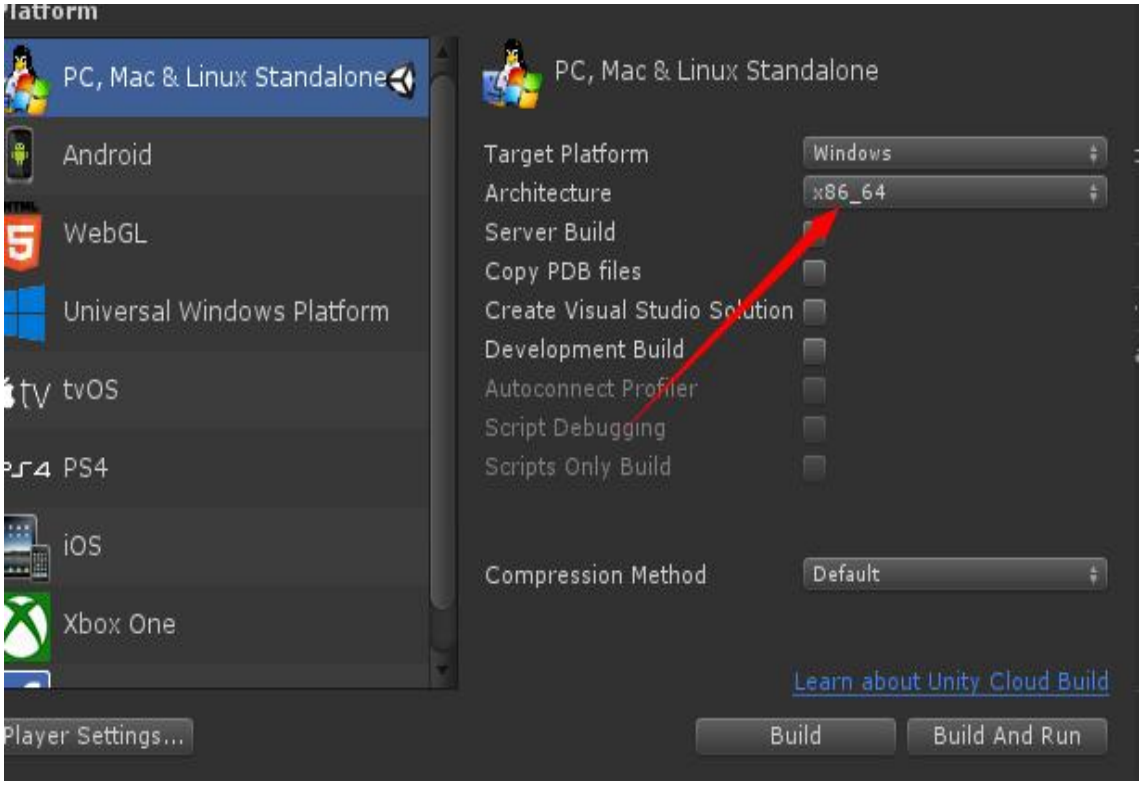
一、插件导入：

1.1 适用于 Unity2019.4 以及 2020 全部版本。

1.2 将插件 EVIS 3D SDK_URP_V1.03unitypackage 直接拖入 unity 面板后导入。或者通过 Assets/Impot package/Custom/Package 选择 EVIS 3D SDK_URP_V1.03.unitypackage 导入。



1.3 设置 unity 发布框架为 X86_64,否则光笔在发布后将无法启动（File/BulidSetting）



二、插件文件内容：

Code：插件脚本

Document：使用文档

Demo：Demo 所有模型材质贴图

Mode：光笔模型，材质球

Plugins：光笔摄像机 dll

Prefabs：3D 立体摄像机预制体

Scenes ：事例 demo

三、插件使用：

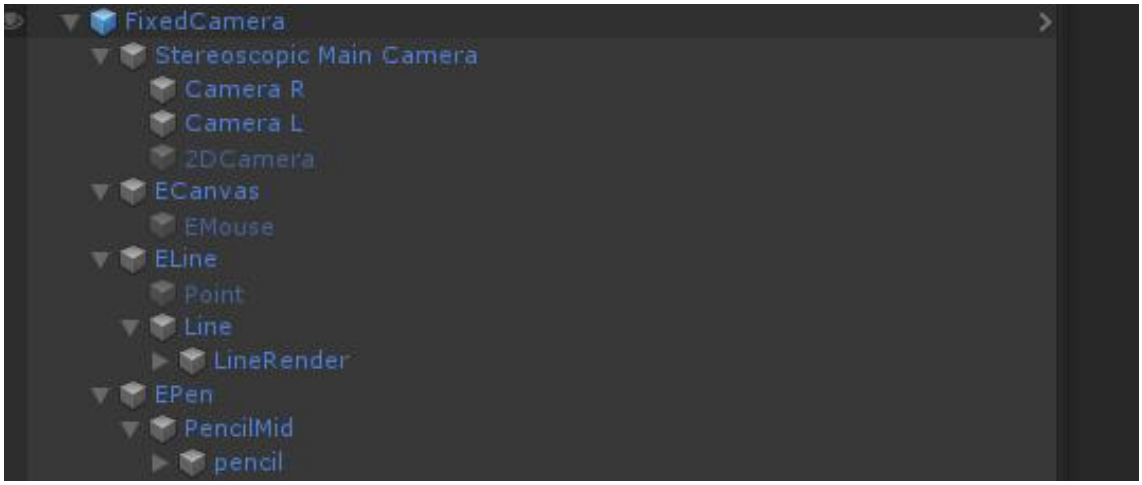
预制体立体效果设定的参数，为最佳实践参数，一般不建议修改。

展示类 3D 摄像机预制体 Fov25 找到插件预制体

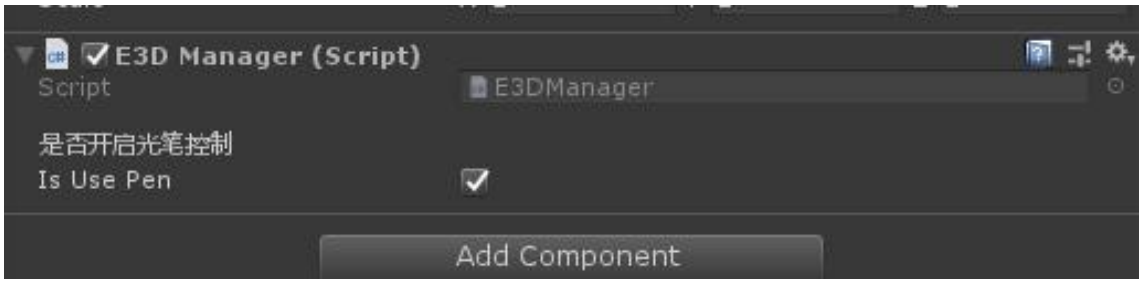
Assets/Prefabs/25Camera/FixedCamera。直接拖入场景中。

新添 unity EventSystem 组件 否则 UI 事件无法触发

3.1 立体相机结构



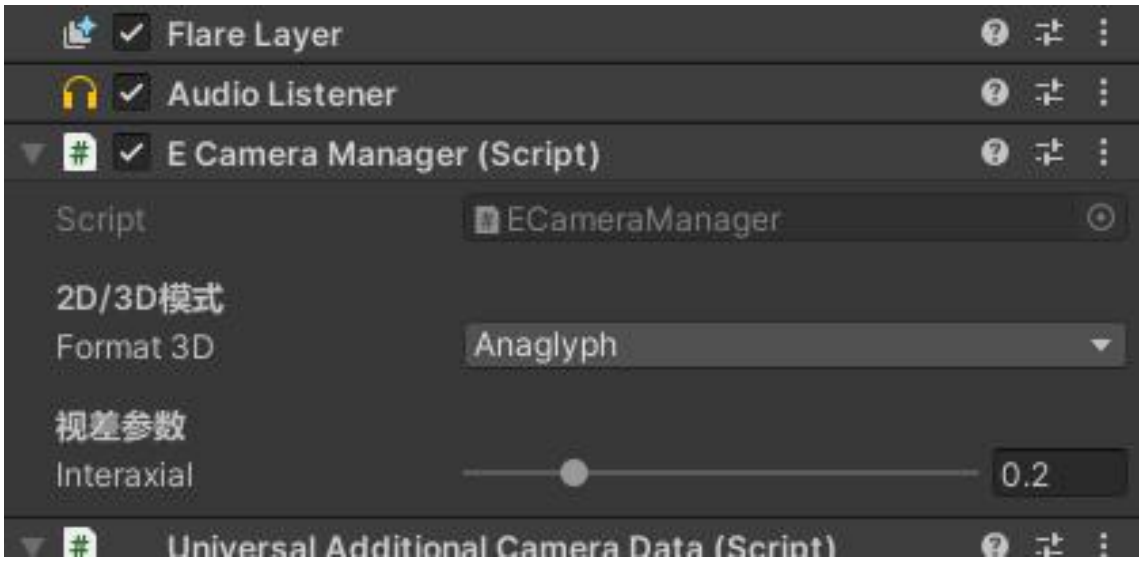
3.2 结构 FixeCamera:



FixeCamera 存在脚本 E3DManager: 是立体摄像机全局控制器。公开的属性是是否使用光笔的属性值，勾选后，启动光笔交互系统，不勾选时候启动鼠标交互（鼠标交互和 Unity 交互 UI 和 3D 物体一样）

脚本中持有 3D 立体相机中各个管理者：EUIManager（UI 管理类）ECameraManager（立体相机管理类） EEventManager（事件管理类）EPenManager(光笔管理类）ERayManager(射线管理类）EUDPManager（UDP 通讯管理类）。

3.3 结构 ECamera：



ECamera 存在脚本 ECameraManager，面板上的公开属性：

Format3D ：SideBySide(左右格式)、NormalMode（单屏格式）

Interaxial：瞳距值，控制 2 个左右渲染摄像机的间距，间距越大，视差越大，立体效果越强烈。

在 Scene 场景中有立体效果的提示框，红绿黄，当物体在红绿之间时，物体呈现出入屏状态；当物体在绿色框位置时，物体呈现和普通 2D 效果一样，无立体效果（一般建议 UI 放在此处）；当在绿色框和黄色框之间时候，物体呈现出屏效果状态。如下图 3.1：

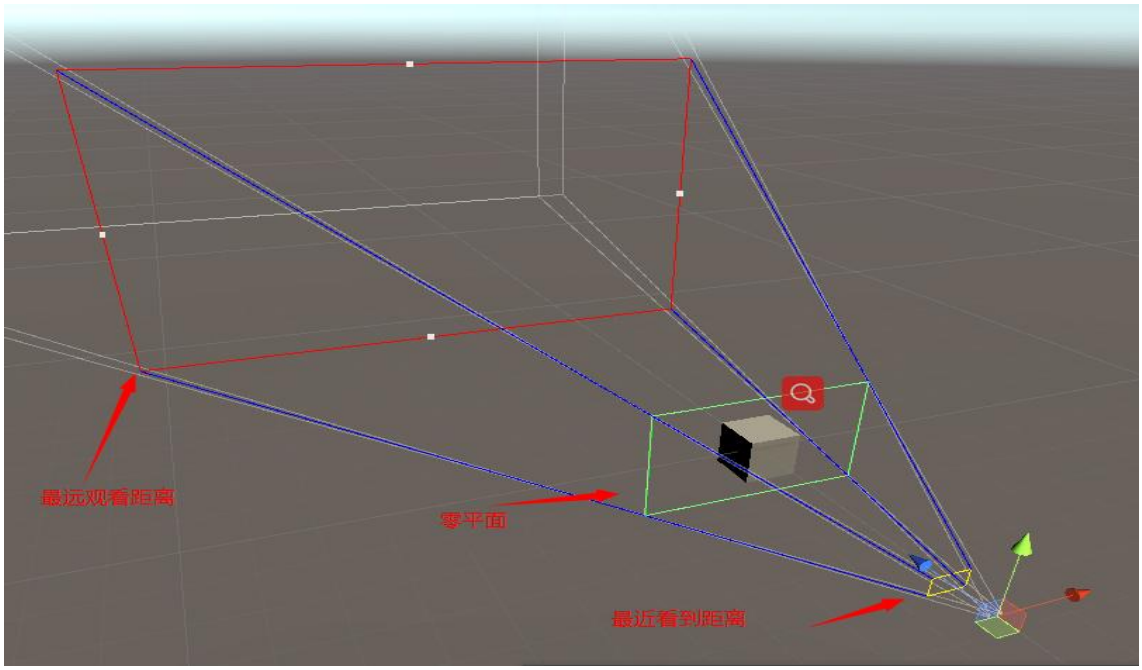
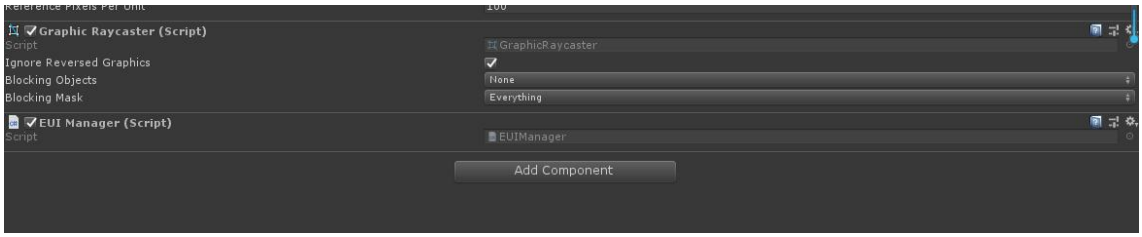


图 3.1

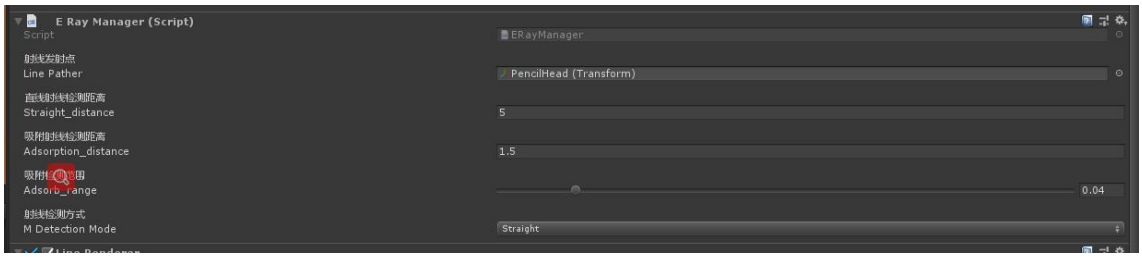
3.4 结构 ECanvas：

ECanvas 存在脚本 EUIManager,脚本功能为立体鼠标的逻辑控制，包括移动，限制，立体鼠标模拟真实鼠标事件，能适配 unity 内置所有的鼠标事件。推荐 UI 在该结构下进行设计，EMouse 虚拟鼠标



3.5 结构 ELine

Eline 存在脚本 ERayManager，脚本功能为射线渲染生成，射线检测功能逻辑。



面板公开属性

LinePather :射线发射点

Straight_distance:射线检测距离

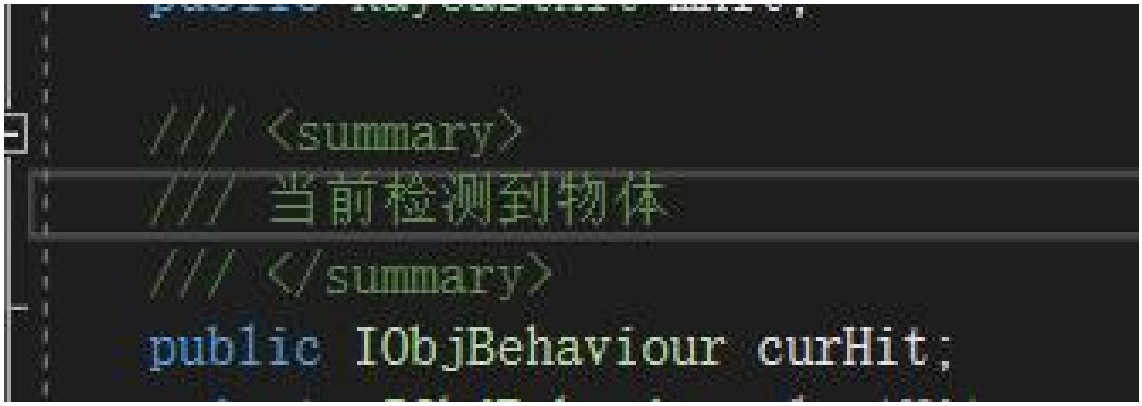
Adsorption_distance:吸附射线检测距离

Adsorb_range:吸附检测范围

mDetectionMode:射线检测方式 直线 吸附

Point：射线特效放置点

在此脚本中，可以获取当前射线检测到的物体



3.6 结构 EPen

EPen 存在脚本 EPenManager ，脚本功能为光笔的姿态映射 ，光笔震动，光笔位置限制，光笔速率调节，光笔按键事件。

面板公开属性：

Offset_y ：光笔 Y 轴方向偏移(局部坐标)，单位米 。

Offset_z:光笔 Z 轴方向偏移(局部坐标)，单位米 。

Offset_x:光笔 x 轴方向偏移(局部坐标)，单位米 。

Offset_xy_rate:光笔 xy 轴方向和真实光笔位移比率（例：现实中光笔位移 1，unity 位移 6）

Offset_z_rate:光笔 Z 轴方向和真实光笔位移比率（例：现实中光笔位移 1，unity 位移 8）

Offset_Z_Limit:光笔 Z 轴局部坐标限制，单位米，（例：-2，说明光笔 Y 轴移动范围为-2 到无穷

光笔事件：

光笔按键委托：（index 对应三个按键：1 为中键，2 为左键，3 位右键）

```
public delegate void PenClickActionHandeler(int index);
/// <summary>
```

光笔点击事件：（index 对应三个按键：1 为中键，2 为左键，3 位右键）

```
/// <summary>
/// 当光笔点击
/// </summary>
public event PenClickActionHandeler OnPenButtonClick;
/// <summary>
```

光笔持续按下事件：（index 对应三个按键：1 为中键，2 为左键，3 位右键）

```
/// <summary>
/// 当光笔按下
/// </summary>
public event PenClickActionHandeler OnPenButtonDown;
/// <summary>
```

光笔抬起事件：（index 对应三个按键：1 为中键，2 为左键，3 位右键）

```
/// <summary>
/// 当光笔抬起
/// </summary>
public event PenClickActionHandeler OnPenButtonUp;
```

光笔接口：

光笔震动接口：

```
/// <summary>
/// 光笔震动
/// </summary>
/// <param name="motorCtrl"></param>
0 个引用
public void Pluse(PenSDK.MotorCtrlType motorCtrl)
{
    penSDK.PenVibrate(motorCtrl);
}
```

光笔震动参数 MotorCtrlType

VIBRATE_OFF = 0, （关闭振动）

VIBRATE_ALLWAYS, （持续一直振动 （如果需要停止振动，发送 VIBRATE_OFF））

VIBRATE_ONCE, （振动一次）

VIBRATE_TWICE, （连续振动两次）

VIBRATE_THRICE, （连续振动三次）

VIBRATE_ONCE_LOOP, （持续间隔的振动（如果需要停止振动，发送 VIBRATE_OFF））

接收光笔数据结构:

```
/// </summary>
private PenSDK.VIDataStructure_t s_RecvMotionStruct = new PenSDK.VIDataStructure_t();
#endregion
```

public ushort frameIndex; (帧序号 0~65535)

public byte freq; (数据频率)

public float[] quat; (光笔四元数 w, x, y, z)

public float[] position; (光笔坐标位置 x,y,z)

public float[] positionHead; (光笔坐标位置 x,y,z)

public int[] key; (按键状态：1 表示按下，0 表示松开)

public byte penPosiStatus; (光笔的位置状态：即是否检测到画面中有光笔： 1，检测到画面有光笔；0，光笔处于外面之外)

四、其他接口说明

4.1EDPManager 脚本：

StartConnect():连接立体显示屏

StopConnect() ：断开链接立体显示屏

```
/// <summary>
/// 停止连接人眼跟踪设备
/// </summary>
0 个引用
public void StopConnect() {
```

EyeTrackingToCamera()：用于人眼跟踪坐标数据和立体相机位置做匹配，模拟人物头部位移和 Unity 立体相机做匹配。

```
/// <summary>
/// 人眼跟踪双眼位置匹配虚拟摄像机
/// </summary>
0 个引用
public void EyeTrackingToCamera(Camera mCa)
{
```

SetCamera() ：控制立体显示屏合成立体图像顺序

```
/// <summary>
/// 控制硬件显示左右画面顺序
/// </summary>
/// <param name="isLeft"></param>
0 个引用
public void SetCamera(bool isLeft) {
```

SetCameraDepth()：设置摄像机景深程度，范围为-128 至 128

```
/// <summary>
/// num >=-128 <=128 设置景深程度
/// </summary>
/// <param name="num"></param>
0 个引用
public void SetCameraDepth(int num) {
```

GetEyeTrackOrginData()：获得人眼跟踪原始 json 数据

```
/// <summary>
/// 获得人眼跟踪原始数据
/// </summary>
/// <returns></returns>
4 个引用
public string GetEyeTrackOrginData() {
```

GetEyeDistance():获得双眼瞳距

```
/// <summary>
/// 得到人眼瞳距
/// </summary>
/// <returns></returns>
0 个引用
public float GetEyeDistance() {
```

GetEyeMiddle()：获得双眼中点

```
/// <summary>
/// 得到人眼中点
/// </summary>
/// <returns></returns>
0 个引用
public Vector3 GetEyeMiddle() {
```

GetLeftEyePostion()：获得左眼坐标（目前坐标左右眼 Z 轴数据一致）

```
/// <summary>
/// 得到左眼坐标
/// </summary>
/// <returns></returns>
0 个引用
public Vector3 GetLeftEyePostion() {
```

GetRightEyePostion() 获得右眼坐标（目前坐标左右眼 Z 轴数据一致）

```
/// 得到右眼坐标
/// </summary>
/// <returns></returns>
0 个引用
public Vector3 GetRightEyePostion()
{
```

SwitchMode(SceneMode mode)：屏幕 2D/3D 切换

SceneMode：

TwoD 2D 模式

ThreeD 3D 模式

```
/// <summary>
///切换硬件屏幕2D 3D显示
/// </summary>
0 个引用
public void SwitchMode(SceneMode mode) {
```

RECT_INFO GetProtectPos() ：获得项目窗口坐标

RECT_INFO 结构：

public int Left　当前矩形范围的最左边界

public int Top;　当前矩形的最上边界

public int Right; 当前矩形的最右边界

public int Bottom; 当前矩形的最下边界

```
/// <summary>
/// 获得窗口坐标范围
/// </summary>
/// <returns></returns>
0 个引用
public RECT_INFO GetProtectPos() {
```

五、PenSDK 脚本：

初始化光笔。

WorldSpace_enum:

WS_Geo,(默认坐标系，右手坐标系（X 轴指向右，Y 轴指向前，Z 轴指向上)

WS_Unity, (Unity 中默认坐标系，左手坐标系（X 轴指向右，Z 轴指向前，Y 轴指向上))

```
[DllImport("VITrackFusionSDK")] // SDK初始化，须指定目的坐标系
1 个引用
private static extern bool SDKInit(WorldSpace_enum space);
```

获得光笔姿态，按键数据：

```
[DllImport("VITrackFusionSDK")] // sdk中数据的回调函数
1 个引用
private static extern void SetDataStructRecvCallback(DataStructRecvCallback callback);
```

光笔报错回调接口：

```
[DllImport("VITrackFusionSDK")] // sdk中数据的回调函数
1 个引用
private static extern void SetDataStructRecvCallback(DataStructRecvCallback callback);
```

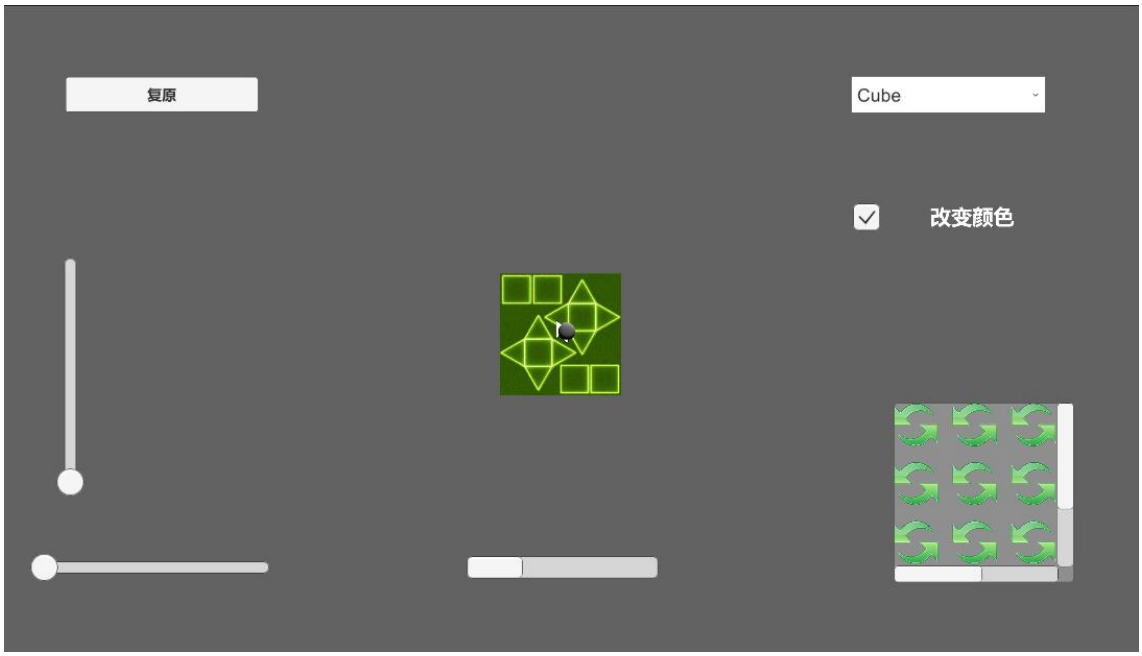
摄像头故障回调接口：

```
[DllImport("VITrackFusionSDK")] // 双目摄像头故障回调函数
1 个引用
private static extern void SetCameraBreakCallback(CameraBreakCallback callback);
```

六、Demo 说明

6.1 UI 交互 Demo

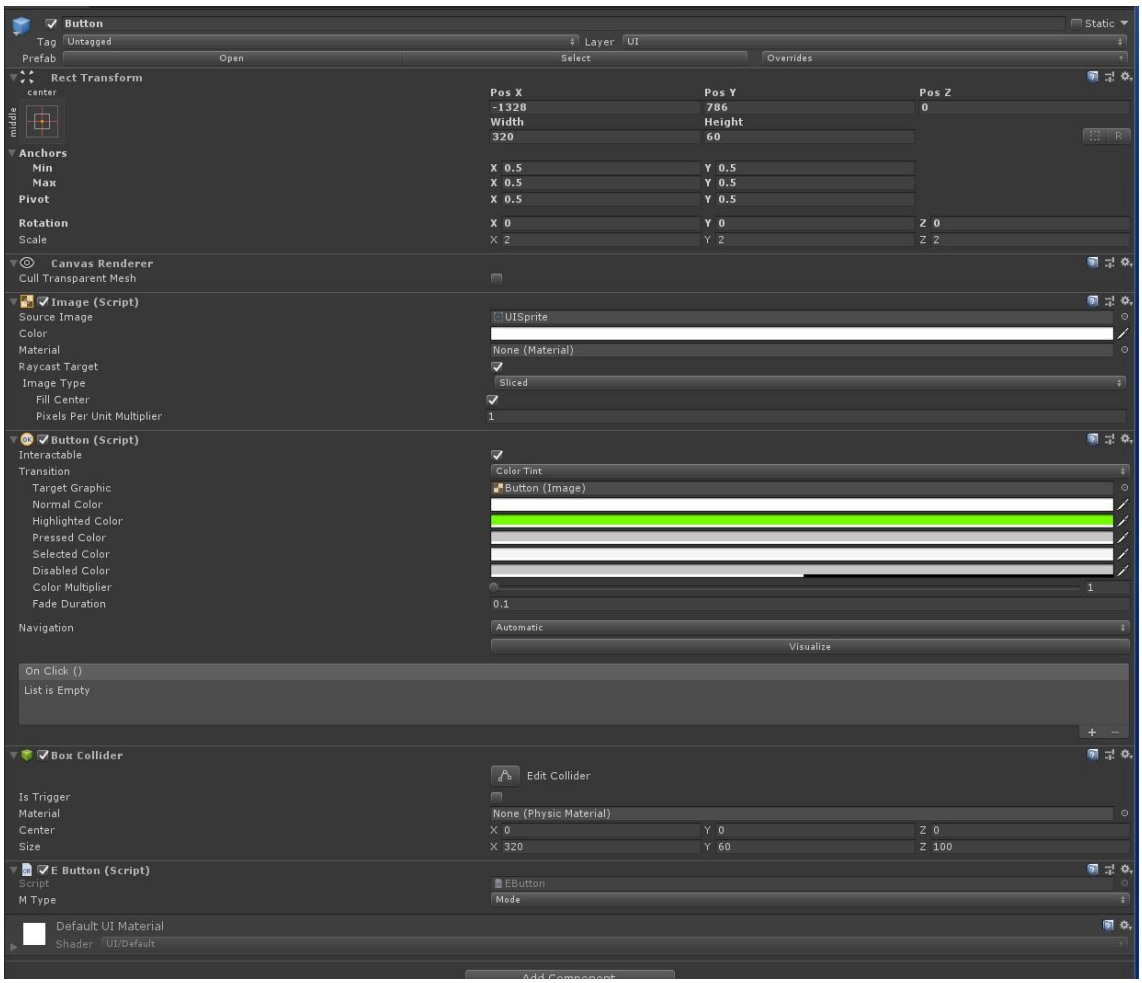
Demo 位置（Assets/Scenes/DemoUI）



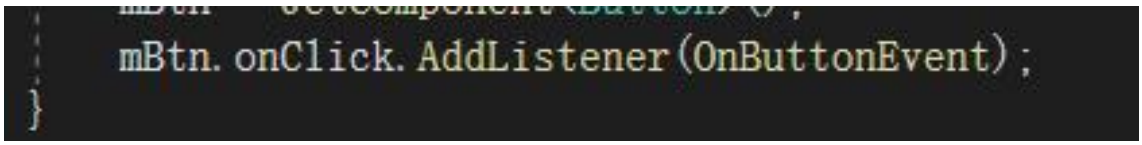
此 Demo 演示了光笔和 UGUI 交互的事例。

适配了 UGUI 的 Button、Toggle、Dropdown、Scrollbar、ScrollView、Slider。分别对应了脚本 EButton、EToggle、EDropdwon、EScrollbar、EScrollView、ESlider.

例如：下图 button 按钮，直接添加 Ebutton 脚本 ，同时为 UI 添加 Collider，就可以和 Button 进行交互，事件的注册和 UGUI 的注册方式一致。



EButton 事件注册：



```
    mBtn.onClick.AddListener(OnButtonEvent);
}
```

Ebutton 事件接口：

光笔选中后按下：

```
/// 1为中键 2 为左建 3为右键
/// </summary>
/// <param name="buttonId"></param>
2 个引用
public void OnPenClick(int buttonId)
{
```

光笔选中后抬起：

```
2 个引用
public void OnPenUp(int buttonId)
{
```

光笔选中后持续按下：

```
2 个引用
public void OnPenDown(int buttonId)
{
```

射线进入按钮：

```
5 个引用
public void RayEnter()
{
```

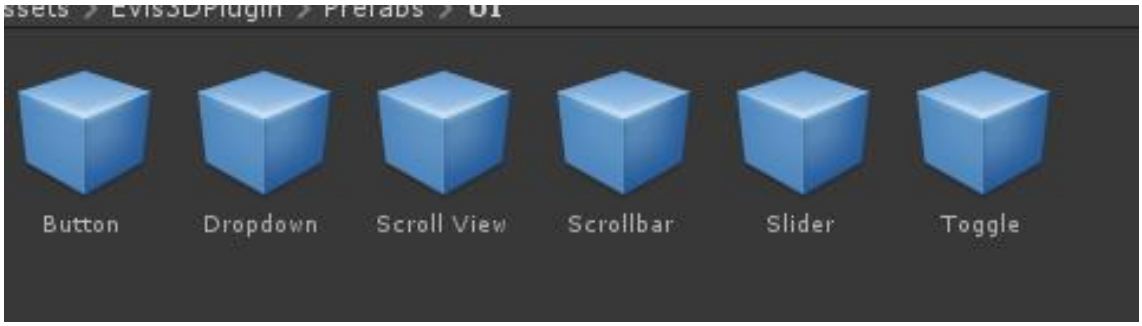
射线停留按钮：

```
5 个引用
public void RayExit()
{
```

射线离开按钮：

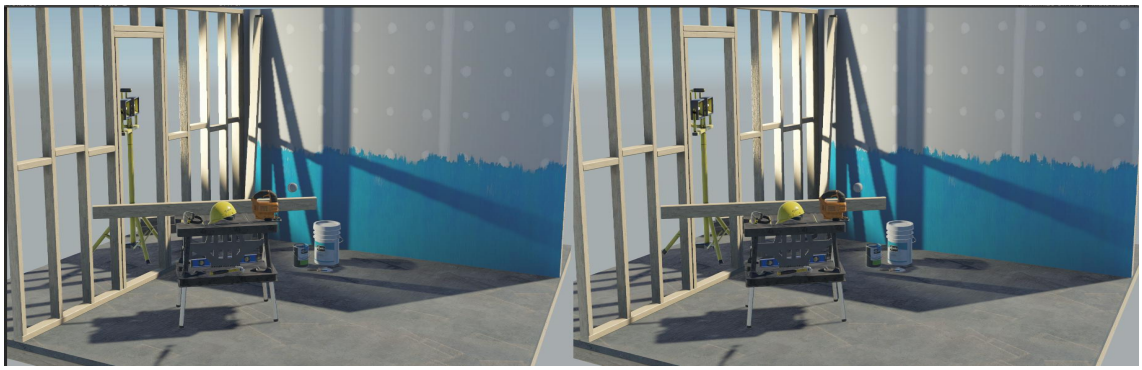
```
3 个引用
public void RayStay()
{
```

同时也预制了 UI 的预制体，可以在 Assets/Evis3DPlugin/Prefabs/UI 中找到

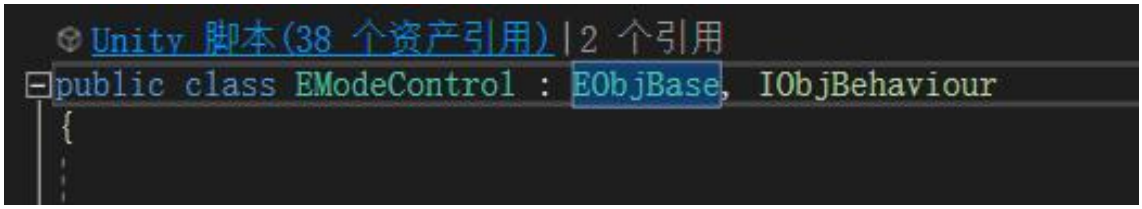


七、3D 交互 Demo

Demo 演示了光笔和 3D 物体的交互，包括拾取，放下，持有以及当光笔射线进入、离开等的高亮提示。演示了 3D 物体的坐标限制。



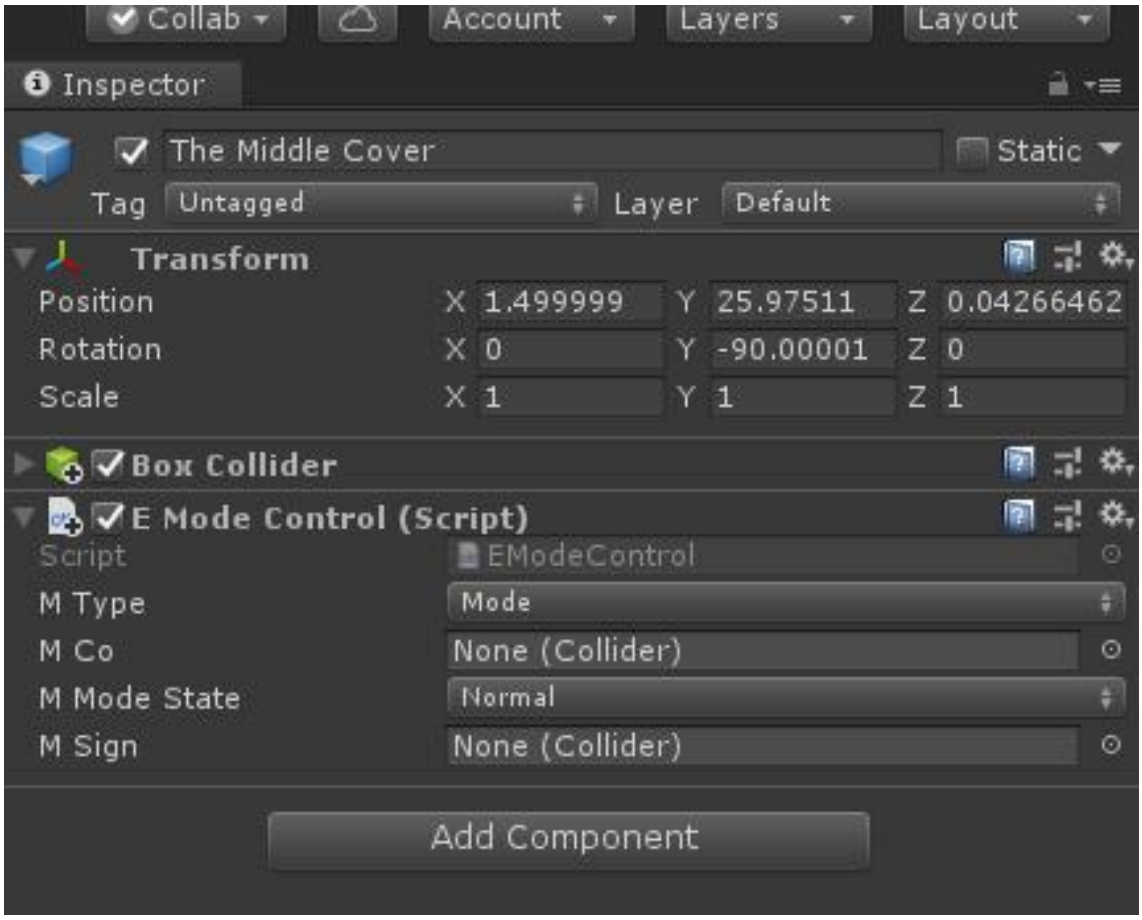
开发中可以自行创建脚本，同时继承父类 EObjBase，和接口 IObjBehaviour。



在接口 IObjBehaviour 中实现如下方法

```
public interface IObjBehaviour
{
    /// <summary>
    /// 射线进入
    /// </summary>

    void RayEnter();
    /// <summary>
    /// 射线停留
    /// </summary>

    void RayStay();
    /// <summary>
    /// 射线离开
    /// </summary>

    void RayExit();

/// <summary>
/// 光笔点击
/// </summary>
/// <param name="buttonId"></param>

  void OnPenClick(int buttonId);
/// <summary>
/// 光笔按下
/// </summary>
/// <param name="buttonId"></param>

  void OnPenDown(int buttonId);
/// <summary>
/// 光笔抬起
/// </summary>
/// <param name="buttonId"></param>

  void OnPenUp(int buttonId);
}
```

同时为需要触发事件的 3D 物体，加上 collider。这样光笔就能实现和 3D 物体的交互。



1.02 更新内容

1. 新增了新的演示 Dem 场景 DemoMultiScreen。

2. 新增了立体摄像机预制体 FixedMultiScreenCamera

3. 修改了 FixedMultiScreenCamera 的零平面到 1

4. 修改了 FixedMultiScreenCamera 的视差到 0.05

5. 修改了 FixedMultiScreenCamera 的中心点和主摄像机一致

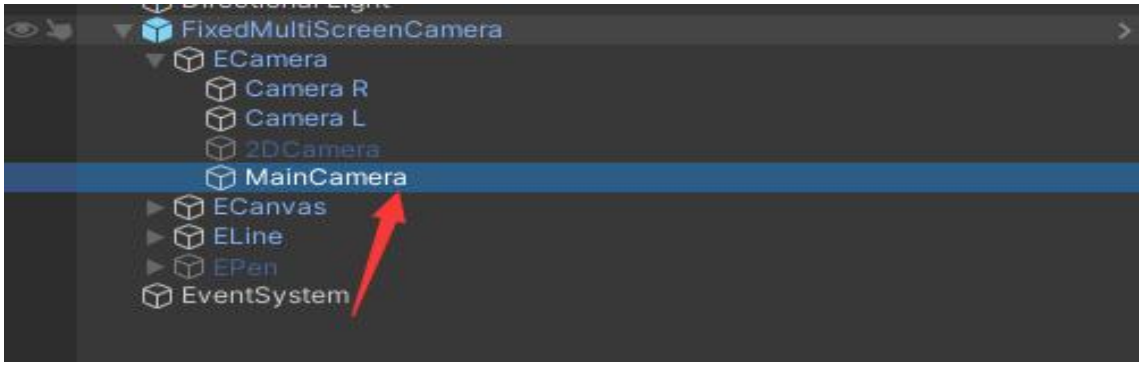
6. 目前仅支持鼠标板的多屏扩展，光笔版本在 2D 显示屏上表现不佳

使用方法：

1. 找到预制体 FixedMultiScreenCamera。路径 Assets/Evis3DPlugin/Pefabs/MultiScreenCamera

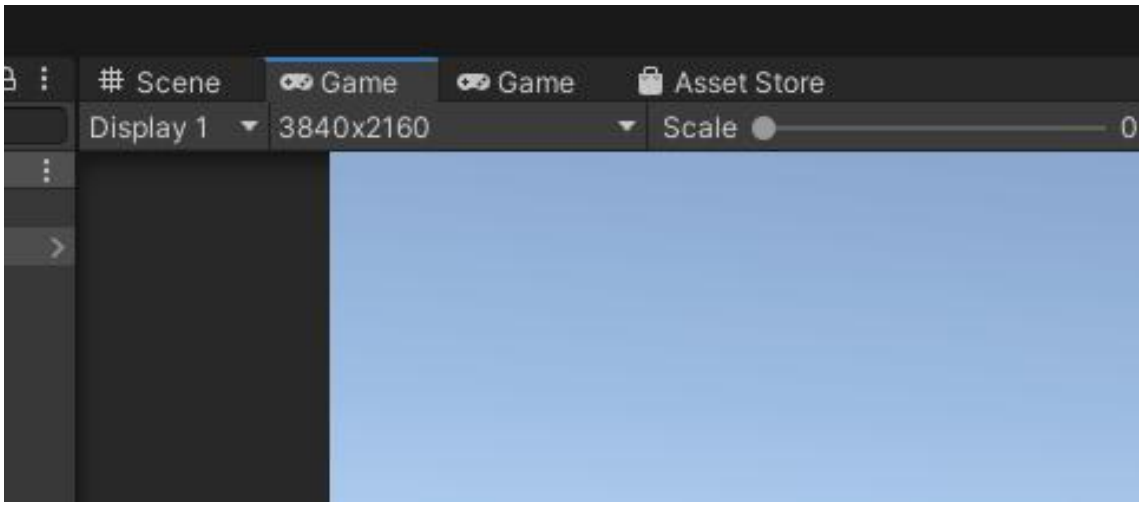
2. 在 hierarchy 面板中添加 EventSystem 组件

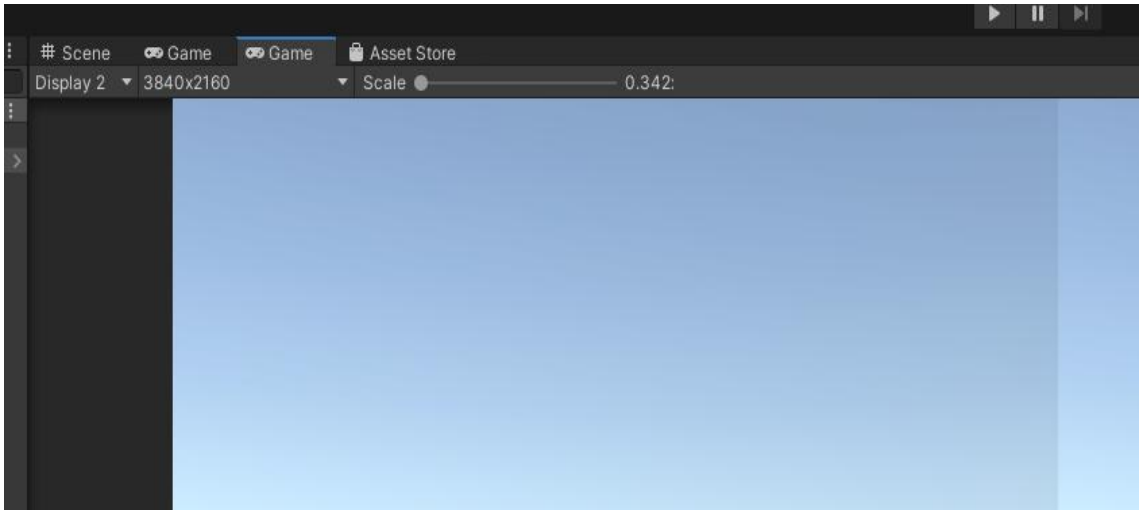
3. MainCamera 为分屏的 2D 主相机，目标屏幕为 DisPlay1，负责控制端的 UI 展示以及 2D 场景展示。



4. CameraR 和 CameraL 和立体相机，目标屏幕为 DisPlay 2 目的为展示立体场景画面

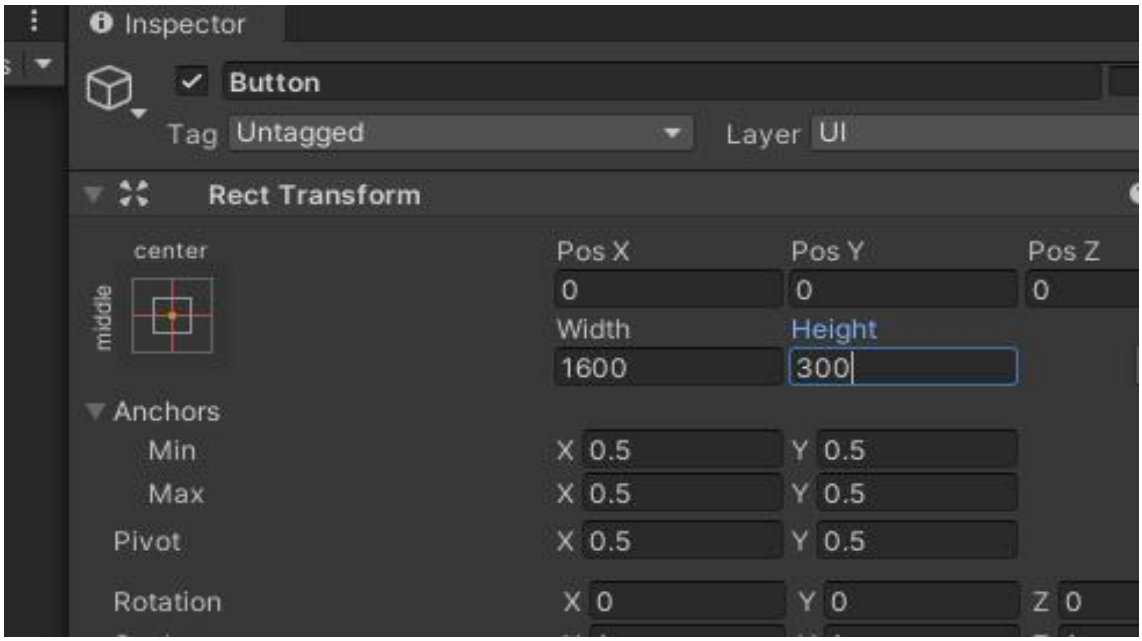
5. 在开发时可以通过新增 Game 视口 ，并改变目标 DisPlay 的方式同时查看 2D 和左右格式屏幕，操作如下图：





注意点：

1.由于立体摄像机影响。Unity 内置的 UI 在主摄像机的分辨率不足，需要将 UI 的分辨率 X10 倍，然后进行适配 ，才能得到清晰的 UI 显示



2. 3D 物体的距离零平面（绿色框框）前后在 0.3 个 unity 单位内 ，太远会出现模糊现象

3. 该程序必须在 2D 显示屏打开，否则会出现 2D 屏幕出现在大屏的现象。