#所有挂载到游戏对象上的脚本中包含的类都继承自MonoBehaviour类

#初始化脚本的代码必须置于Awake() 或Start() 方法中。不同之处在于 :

Awake() 方法是在加载场景时运行，在所有Start() 方法之前运行

Start() 方法是在第一次调用Update() 或FixedUpdate() 方法之前调用

#\*Unity脚本中协同程序(Coroutines)必须是IEnumerator返回类型，并且yield用yield return替代。如yield return new WaitForSeconds(2); //等待2s

#\*只有序列化的成员变量才能显示在属性查看器中，而private个protected类型的成员变量只能在专家模式中显示，而且，其属性不被序列化或显示在属性查看器，如果属性想在属性查看器中显示，必须是public的

#不要在构造函数中初始化任何变量，而是使用Awake() 和Start() 方法来实现。并且在单一模式下使用构造函数可能会引发类似随机的空引用异常

#Update()方法都是按帧为时间单位计算的

#FixedUpdate()方法是按固定的物理时间被系统回调执行的

#物体旋转是通过transform.Rotate()方法来实现的

#物体移动是通过transform.Translate()方法来实现的

#Time.deltaTime指的是从最近一次调用Update或者FixedUpdate方法到现在的时间，如果想均匀旋转一个物体，不考虑帧速率的情况下，可以使用

#可通过GetComponent<>()来获取一个对象组件

#transform.Find()来找到对象进行操作

#GameObject.FindWithTag().transform.Rotate()来通过tag找到对象 也可以同理操作上面

#FindObjectOfType<>()可以找到第一个有<>的组件 最后添加最先找到

#Instantiate()来初始化游戏对象

#可通过StartCoroutine()方法启动一个协同程序，所有的协同程序都是在主线程中运行的，还是一个单线程程序，StopAllCoroutine()终止所有可以终止的协同程序

#通过私有变量储存组件，使得程序不会在每一帧查找所需组件

#规定左上角坐标为屏幕原点

#绘制按钮需要重写OnGUI()

#普通文字按钮if (GUI.Button (new Rect (), "文字"))

#带有2D纹理的按钮if (GUI.Button (new Rect (), Texture))

#同理还有GUIContent和GUIStyle按钮，GUIStyle多一个按钮文字变量，if为真为按下后的回调执行

#Screen方法里有屏幕长度和宽度

#GUI.DrawTexture()用于绘制纹理，其中参数scaleMode有StretchToFill会对图片进行拉伸，使之沾满整个矩形。ScaleAndCrop将图片等比缩放充满矩形，多余会被裁剪。ScaleTofit将图片等比缩放使图片完全显示在矩形内

static function DrawTexture (position, image, scaleMode, alphaBlend, imageAspect) : void

alpgaBlend : 图片的混合模式，是否通道混合图片显示，默认为混合通道

imageAspect : 源图片的长宽比，如果为0，则使用图像的长宽比，通过宽/高获得所需的长宽比

#GUIStyle控件中的Alignment : 控件中文本的位置，Word Wrap : 勾选后控件中的文字会被限制在控件的矩形区域内，超出部分会换行