## UI界面

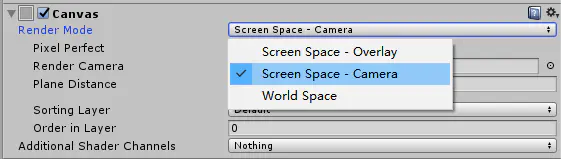
### Canvas画布

Canvas是画布，是摆放所有UI元素的区域，在场景中创建的所有控件都会自动变为Canvas游戏对象的子对象，若场景中没有画布，在创建控件时会自动创建画布。

创建画布有两种方式：一是通过菜单直接创建；二是直接创建一个UI组件时自动创建一个容纳该组件的画布。

不管用哪种方式创建画布，系统都会自动创建一个名为 EventSystem 的游戏对象，上面挂载了若干与事件监听相关的组件可供设置。

在画布上有一个 Render Mode 属性，它有 3 个选项，分别对应画布的三种渲染模式：Screen Space-Overlay、Screen Space-Camera和World Space，如下图所示。

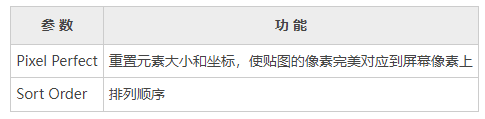


**图1.1 Render Mode 属性**

* **Screen Space-Overlay 渲染模式**

在 Screen Space-Overlay 渲染模式下，场景中的 UI 被渲染到屏幕上，如果屏幕大小改变了或更改了分辨率，画布将自动更改大小以适配屏幕。此种模式不需要 UI 摄像机，UI 将永远出现在所有摄像机的最前面。

Screen Space-Overlay 渲染模式的参数如下图所示：



**图1.2 Overlay 渲染模式参数**

* **Screen Space-Camera 渲染模式**

Screen Space-Camera 渲染模式类似于 Screen Space-Overlay 渲染模式。这种渲染模式下，画布被放置在指定摄像机前的一个给定距离上，它支持在 UI 前方显示 3D 模型与粒子系统等内容，通过指定的摄像机 UI 被呈现出来，如果屏幕大小改变或更改了分辨率，画布将自动更改大小以适配屏幕。

Screen Space-Camera 渲染模式的参数如下图所示：



**图1.3 Camera渲染模式参数**

* **World Space渲染模式**

在 World Space 渲染模式下呈现的 UI 好像是 3D 场景中的一个 Plane 对象。与前两种渲染模式不同，其屏幕的大小将取决于拍摄的角度和相机的距离。它是一个完全三维的 UI，也就是把 UI 也当成三维对象，例如摄像机离 UI 远了，其显示就会变小，近了就会变大。

World Space 渲染模式的参数如下图所示：

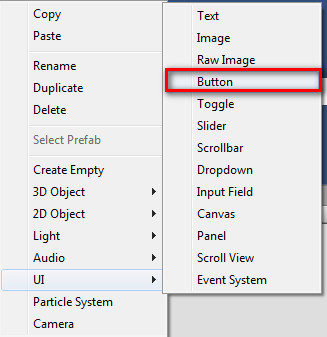


**图1.3** World Space**渲染模式参数**

## UGUI组件

### Button

**（1）新建一个按钮**



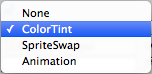
**图1.4 添加Button步骤一**



**图1.5 添加Button步骤二**

**（2）参数介绍**

* Interactable ：该组件是否可点击(设置为false时，将禁用交互，并且过渡状态将设置为禁用状态)；
* Transition:组件的几种过渡效果：



**图1.6 组件过渡效果**

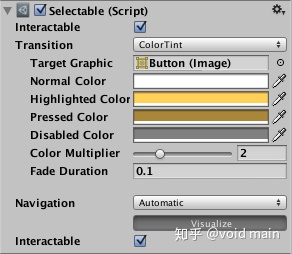
None：此选项使按钮完全没有状态影响。

ColorTint：根据按钮的状态更改按钮的颜色。

SpriteSwap：根据按钮当前所处的状态，允许显示不同的精灵，可以自定义 精灵。

Animation：根据按钮的状态允许发生动画，必须存在一个动画师组件才能 使用动画过渡。确保禁用根运动很重要。要创建动画控制器，请 单击“生成动画”（或创建自己的动画），并确保已将动画控制器 添加到按钮的动画器组件

* Transition-ColorTint:



**图1.7 ColorTint选项**

Target Graphic：用于交互组件的图形；

Normal Color：控件的正常颜色；

Highlighted Color：控件突出显示时的颜色；

Pressed Color：按下控件时的颜色；

Disabled Color：控件禁用时的颜色；

Color Multiplier：这会将每个过渡的色调颜色乘以其值。使用此功能，您可 以创建大于1的颜色，以使基色小于白色（或小于全alpha） 的图形元素上的颜色（或alpha通道）变亮；

Fade Duraction：从一种状态消失到另一种状态所花费的时间（以秒为单位）。

* Transition-SpriteSwap:

Target Graphic：用于交互组件的图形；

Normal Color：控件的正常颜色；

Highlighted Color：控件突出显示时的颜色；

Pressed Color：按下控件时的颜色；



**图1.8 SpriteSwap选项**

Disabled Color：控件禁用时的颜色。

* Transition-Animation:



**图1.9** Animation**选项**

Normal Trigger：普通动画触发器使用；

Highlighted Trigger：突出显示控件时触发使用；

Pressed Trigger：按下控件时触发使用；

Disabled Trigger：禁用控件时触发使用。

**（3）添加点击事件**

创建脚本如下：

using UnityEngine;

using System.Collections;public class Click : MonoBehaviour {

// Use this for initialization

void Start () {

}

// Update is called once per frame

void Update () {

}

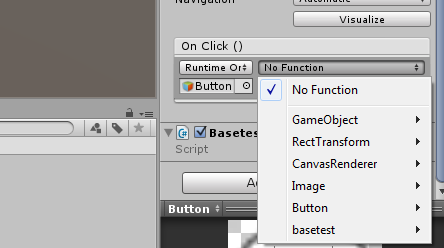
public void OnMyClick() {

Debug.Log("你点击了按钮");

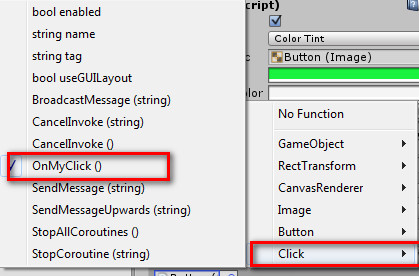
}

}

首先将脚本绑定到按钮上，然后将按钮这个对象绑定要OnClick设置上，具体步骤如下图所示。



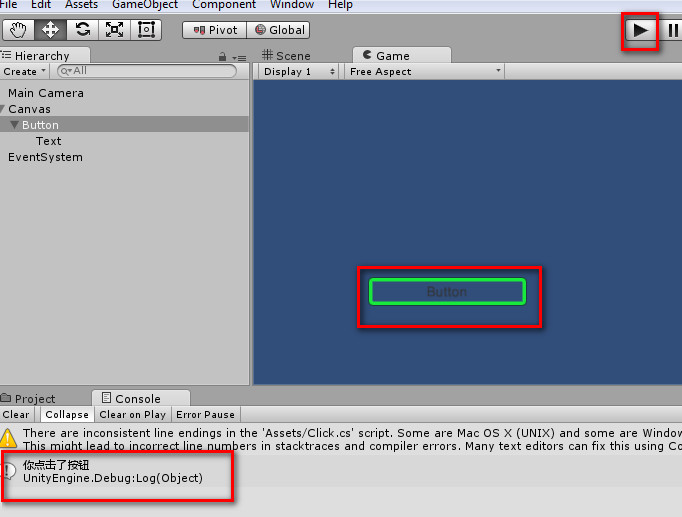
**图1.10 脚本绑定步骤一**



**图1.11 脚本绑定步骤二**

**（4）运行游戏**

运行游戏，点击该按钮，效果如图所示：



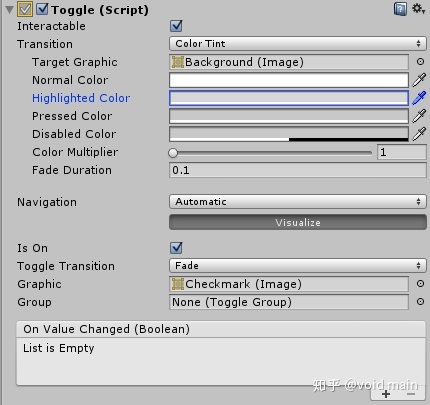
**图1.12 Button组件功能展示**

### Toggle

Toggle 切换控制是一个复选框，允许用户打开或关闭的一个选项。

此组件中的交互、过渡及导航与上文中Button组件中的介绍相同。



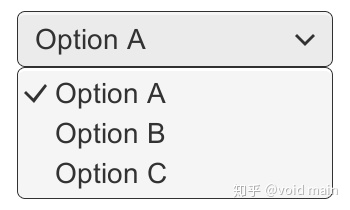


**图1.12 Toggle组件**

### DropDown

该组件可以用来让用户选择选项列表中的一个选项。控件显示当前选择的选项。单击后，它会打开选项列表，因此可以选择一个新选项。选择新选项后，列表将再次关闭，并且控件将显示新选择的选项。如果用户单击控件本身或“画布”中的其他任何位置，该列表也将关闭。

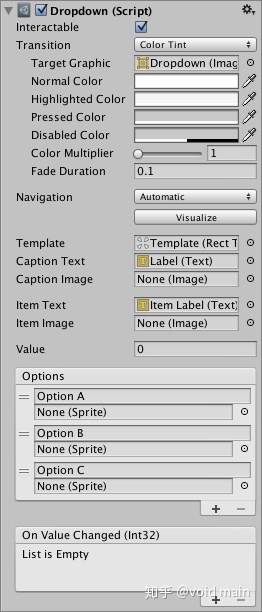




**图1.13 Dropdown组件**

此组件中的交互、过渡及导航与上文中Button组件中的介绍相同。

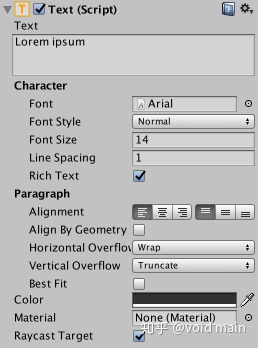
* Template：下拉列表模板的Rect转换；
* Caption Text：文本组件，用于保存当前所选选项的文本；（可选的）
* Caption Image：Image组件，用于保存当前所选选项的图像；（可选的）
* Item Text：用于保存项目文本的Text组件；（可选的）
* Item Image：Image组件，用于保存项目的图像；（可选的）
* Value：当前所选选项的索引。0是第一个选项，1是第二个选项，以此类推。
* Options：可能选项的列表。可以为每个选项指定文本字符串和图像。



**图1.14 Dropdown参数**

### Text、Image

**（1）Text**



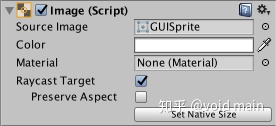
**图1.15 Text组件**

* **Character:**
* Font：字体格式；
* Font Style：Normal（默认），Bold（粗体），Italic（斜体），BoldAndltalic （粗体和斜体）；
* Font Size：字体的大小；
* Line Spacing：文本中的行与行之间的间隔；
* Rich Text：是否使用富文本
* **ParaGraph:**
* Alignment：文本的排版：左对齐，居中，右对齐，

上对齐，居中，下对齐；

* Align By Geometry：几何对齐；
* Horizontal Overflow（水平溢出）：溢出，隐藏溢出；
* Vertically Overflow（垂直溢出）：溢出，截断溢出；
* Best Fit（最佳适配）：设置最大和最小值；
* Color：设置字体的颜色；
* Material：设置字体的材质；
* Raycast Target：是否可点击；

**（2）Image**



**图1.16 Image组件**

* Source Image:设置并且填充需要设置的图像；
* Color：设置图像的颜色；
* Material：设置图像的材质；
* Raycast Target：是否可点击；
* Preserve Aspect：是否保护。