# 操作 9　UGUI

## 9.1 内容概要

- ✧ GUI 图形用户界面
- ✧ 在游戏中添加 GUI 元素
- ✧ 生命值 GUI
- ✧ UGUI 控件
- ✧ UI 动画

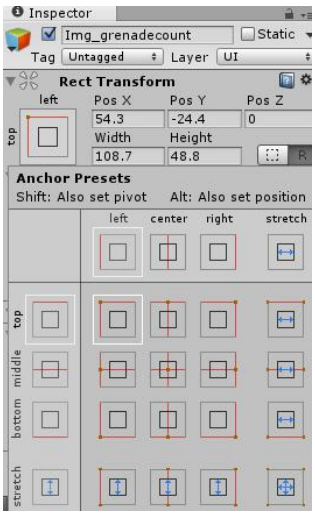
## 9.2 学习目标

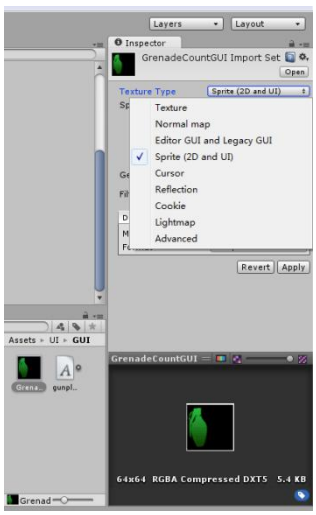
熟练掌握 Unity 中游戏作品 UI 的设计实现原理和方法；熟悉基本控件的应用及设置方法；掌握对话框的应用；熟练掌握 UI 事件的响应方式及应用技巧；熟练掌握 UI 的动画控制方法及技巧。

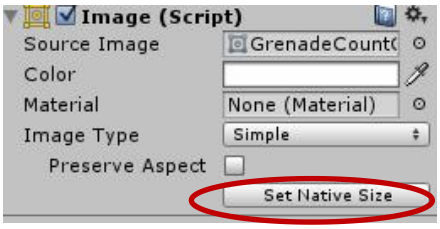
## 9.3 具体内容及实践指导

### 9.3.1 “Widget”范例——右上角显示拾取道具的数量

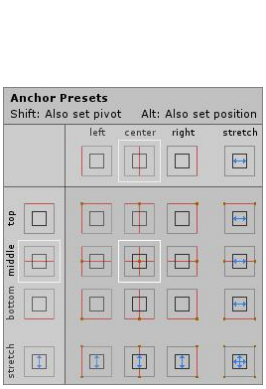

1. 创建 UI——Canvas 和 Image（重命名为 Img_grenadecount），并设置位置锚点（左上），和 Sprite（设置原图大小），如下图：

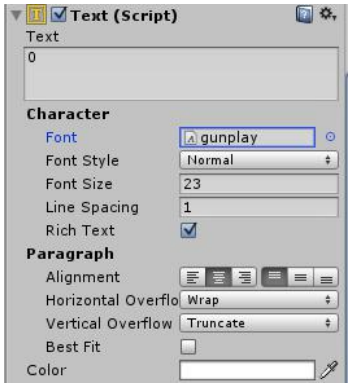



2. 创建 UI——Text，并作为 Img_grenadecount 的子物体，命名为 txt_count，设置 Inspector 中的锚点（调整好位置后，设置锚点中心对齐）字体、居中、大小等，如下图：



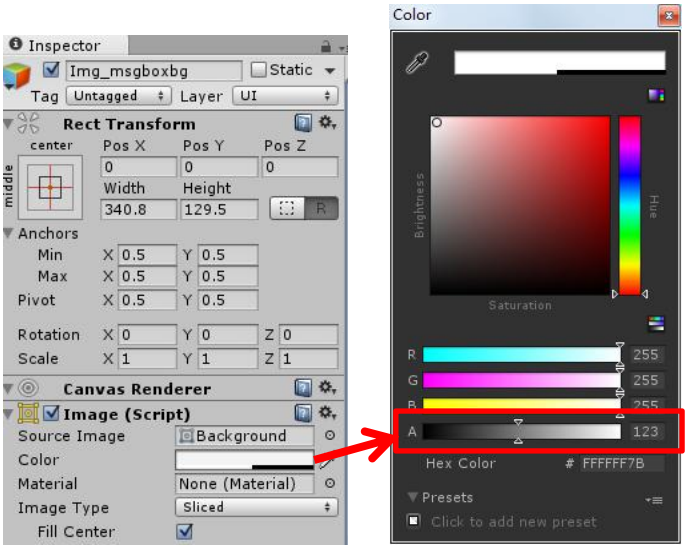

3. 创建脚本 UIControl.cs（或在之前的 GameControl.cs 脚本中修改），添加代码实现玩家碰撞弹药箱后，显示数量增加。

```
using UnityEngine;
using System.Collections;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class UIControl: MonoBehaviour  {
    public GameObject grenadeBox; //赋值弹药箱预制物
    private   int boxNumber = 10;
    static public int GrenadeCount = 0; //静态全局变量里记录手雷的数量
    private Text txtcount;

    // Use this for initialization
    void Start () {
         txtcount = GameObject.Find("txt_count").GetComponent<Text>();
        txtcount.text = GrenadeCount.ToString();
        for (int i = 0; i < boxNumber; i++)
        {
            Vector3 position = new Vector3(Random.Range(-9.0f, 9.0f), 0.5f,
Random.Range(-9.0f, 9.0f));
            Instantiate(grenadeBox, position, Quaternion.identity);
        }
    }
    void OnControllerColliderHit(ControllerColliderHit hit)
    {
        if (hit.gameObject.tag == "grenadeBox")
        {
            Destroy(hit.gameObject);
            UIControl.GrenadeCount += 1;
            //如果是其他脚本中访问静态成员，则通过类名调用
             txtcount.text = UIControl.GrenadeCount.ToString();
        }
    }
```

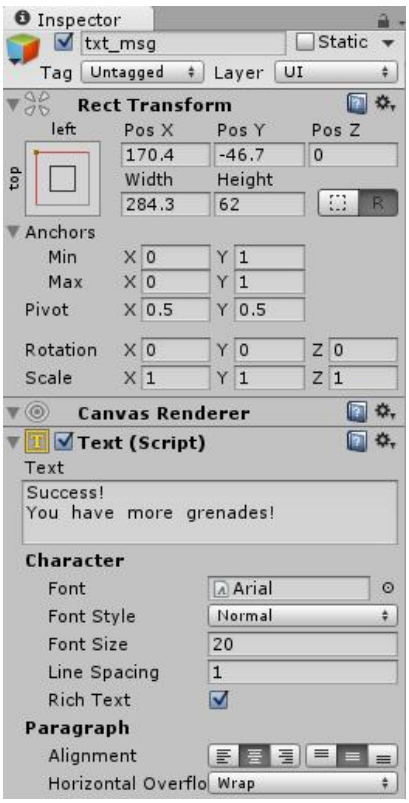
}

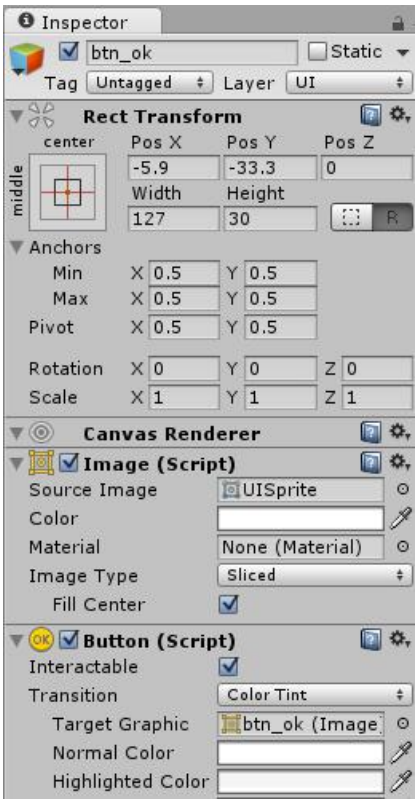
### 9.3.2 “Widget”范例——提示对话框 UI


1. 在 Canvas 下创建 UI——Image（重命名为 Img_msgboxbg），并设置位置锚点（默认居中），和 Sprite（半透明），如下图：

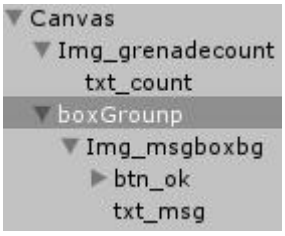



2. 创建 Text 和 Button，分别重命名为 txt_msg 和 btn_ok，并添加为 Img_msgboxbg 的子物体，设置 Inspector，如下图：




3. 创建一个空子物体（canvas 的子物体），命名为 boxGrounp，将 Img_msgboxbg 作为该物体的子物体，如下图：

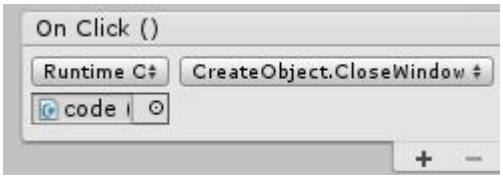



4. 添加脚本（绑定在角色对象上的脚本），初始时不显示该 msgUI，玩家碰撞弹药箱时，显示 UI，如下：

```
private GameObject UImsg;

void Awake()
 {
     txtcount = GameObject.Find("txt_count").GetComponent<Text>();
     UImsg = GameObject.Find("boxGrounp");
 }
 void Start()
 {
     UImsg.SetActive(false);
 }
void OnControllerColliderHit(ControllerColliderHit hit)
    {
        if (hit.gameObject.tag == "grenadeBox")
```

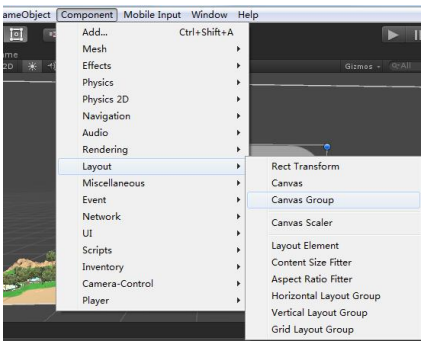
```
        {
            CreateObject.GrenadeCount += 1;
            Destroy(hit.gameObject);
            UImsg.SetActive(true);
            txtcount.text = CreateObject.GrenadeCount.ToString();
        }
}
```

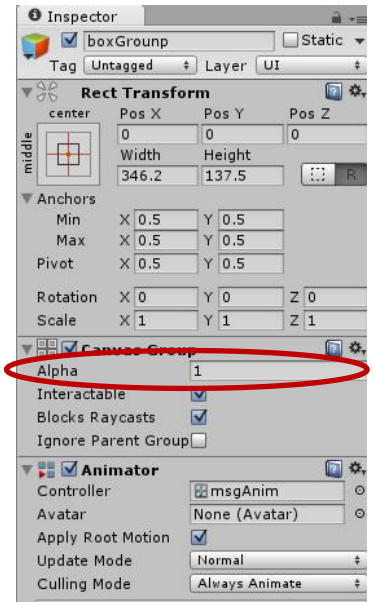
5. 添加脚本，设置按钮事件，实现单击 close 按钮 msgUI 消失，如下：



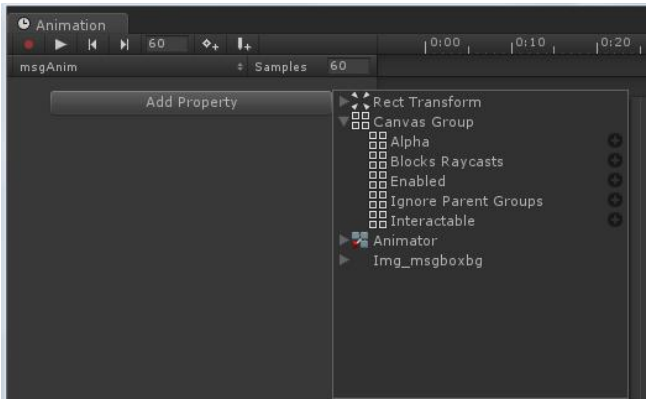

```
public void CloseWindow()
    {
        UImsg.SetActive(false);
    }
```

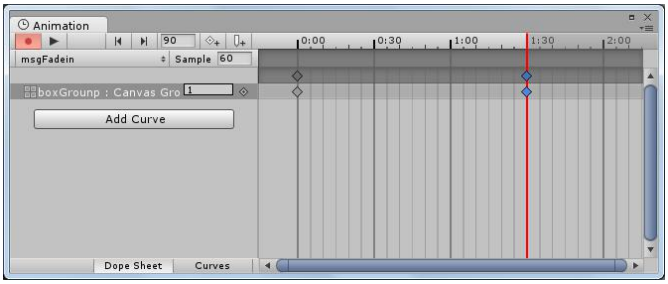
6. 可以给 boxGroup 添加组件 CanvasGroup 和 Animator，增加渐进渐出的动画效果。



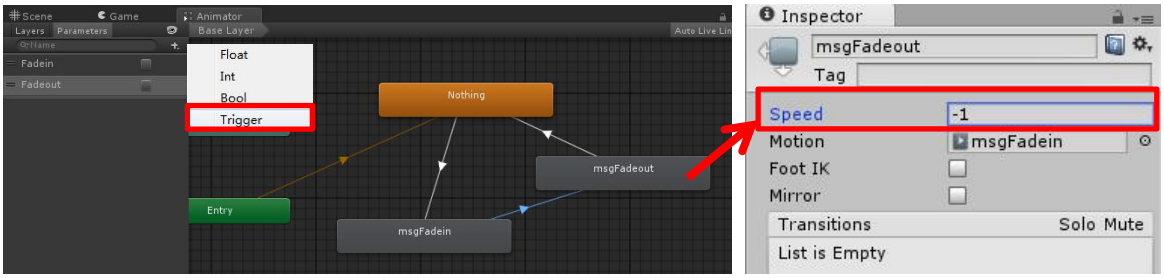



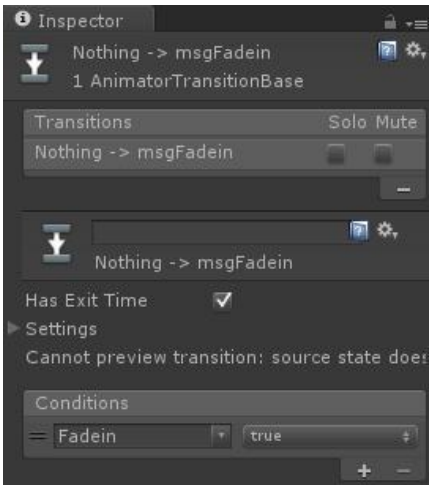

7. 创建 Create——animator controller，命名为 msgAnim，并选中 boxGroup，单开窗口 window——Animation，新建一个动画片段，命名为 msgFadein，设置透明度的值由 0~1，如下图：

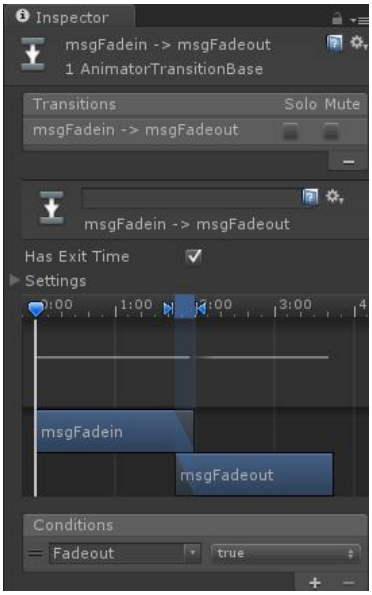



8. 打开控制器 msgAnim，添加动画盒 nothing，和反向动画盒 msgFadeout，添加参数 2 个 Trigger 类型的 paramters：Fadein 和 Fadeout，如下图：





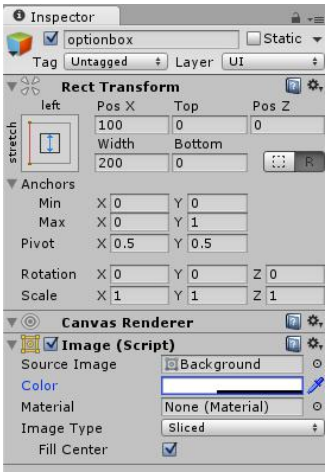
9. 修改脚本，代码如下：

```
private Animator msgAnim;
    // Use this for initialization
void Awake()
    {
        txtcount = GameObject.Find("txt_count").GetComponent<Text>();
        UImsg = GameObject.Find("boxGrounp");
        msgAnim = UImsg.GetComponent<Animator>();
    }
```

```
void OnControllerColliderHit(ControllerColliderHit hit)
    {
        if (hit.gameObject.tag == "grenadeBox")
        {
            CreateObject.GrenadeCount += 1;
            Destroy(hit.gameObject);
            msgAnim.SetTrigger("Fadein");
    //          UImsg.SetActive(true);
            txtcount.text = CreateObject.GrenadeCount.ToString();
        }
}
public void CloseWindow()
    {
        msgAnim.SetTrigger("Fadeout");
    }
```

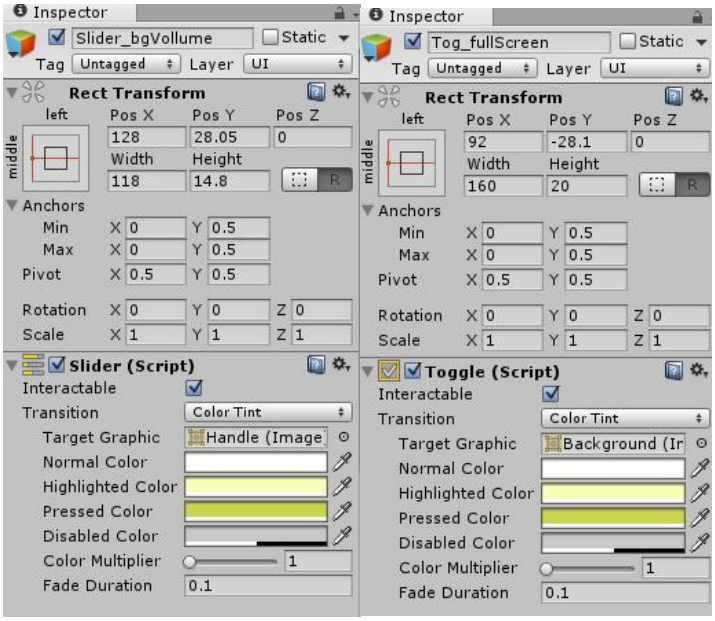
### 9.3.3 “Widget”范例—— “设置”UI

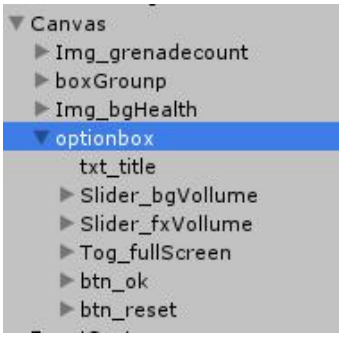

1. 创建 panel，作为 canvas 的子物体，命名为 optionbox 并设置 Inspector（透明度，位置 left 边对齐），如图：



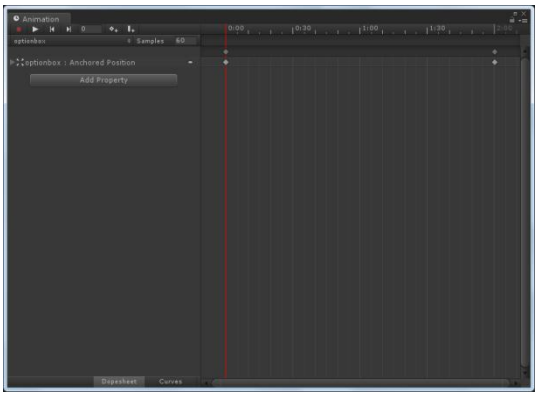

2. 添加 optionbox 子物体：Text(txt_title，参数设置)，Toggle（tog_fullScreen，全屏模式），Slider (Slider_bgVollume,背景音量)，Slider（Slider_fxVollume，特效音量），button(btn_ok，确定；btn_reset，

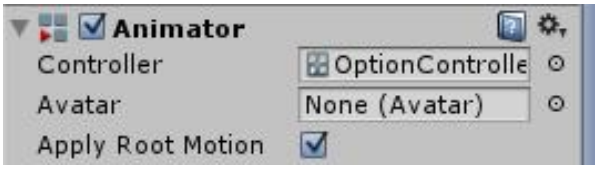
重设)，如图：

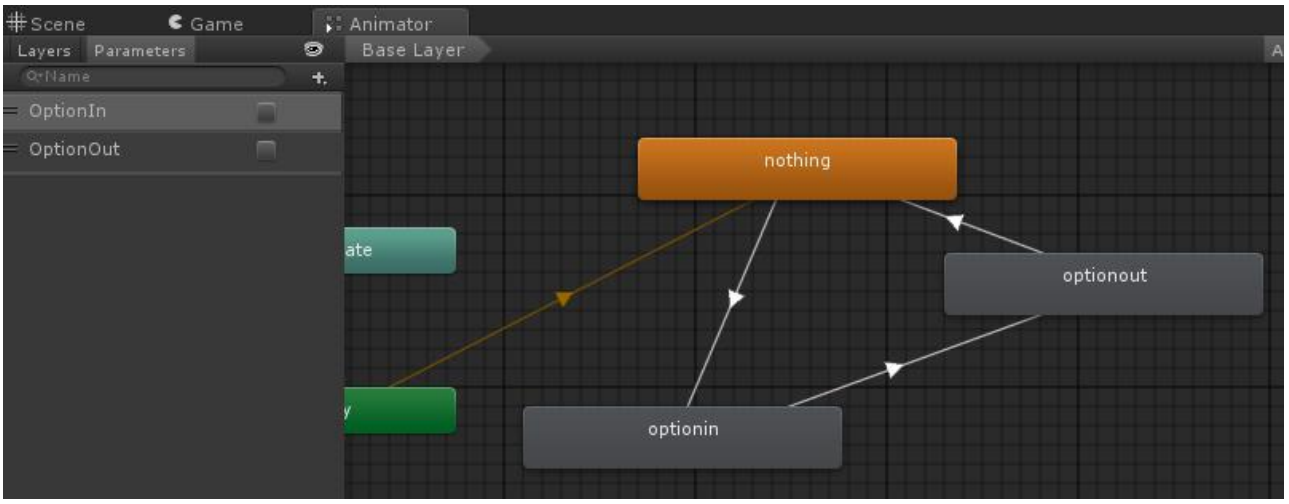


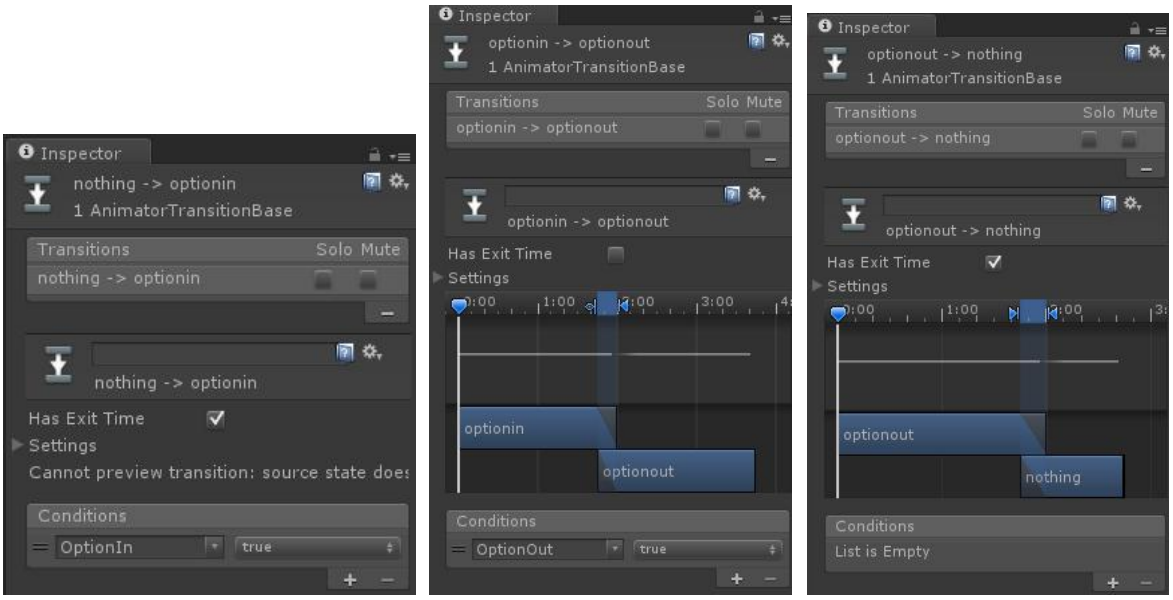




3. 给 optionbox 添加动画组件，并创建动画控制器 OptionController 和动画片段 OptionIn.anim；





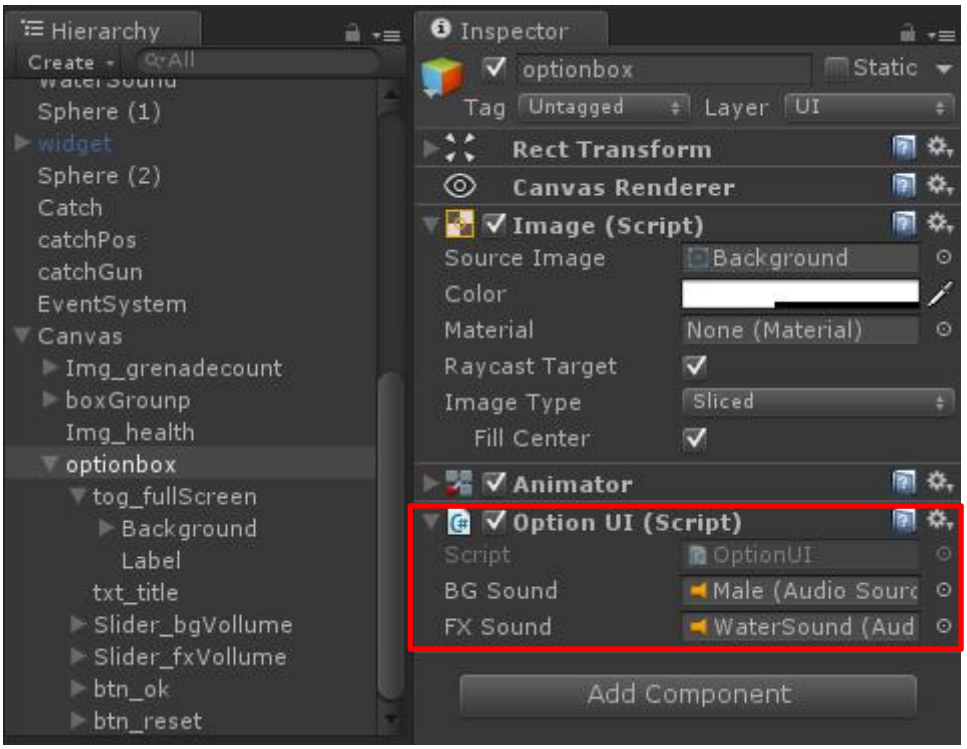



4. 创建脚本 OptionUI.cs，绑定到控件 optionbox 上，并设置按钮、滑块 Slider 和 Toggle 的事件响应函数如下图：

```
public class OptionUI : MonoBehaviour {
    public AudioSource BGSound;//背景音乐声源
    public AudioSource FXSound;//音效音乐声源
    private Slider slider_bgVollume;
    private Slider slider_fxVollume;
    private Toggle tog_fullScreen;
    private Animator optionAnim;
    private bool tagEsc = false;
    private bool tagPause = false;

 void Awake()
 {
     slider_bgVollume = GameObject.Find("Slider_bgVollume").GetComponent<Slider>();
     slider_fxVollume = GameObject.Find("Slider_fxVollume").GetComponent<Slider>();
     tog_fullScreen = GameObject.Find("tog_fullScreen").GetComponent<Toggle>();
     optionAnim = GetComponent<Animator>();
 }
 void Start()
 {
     BGSound.volume = slider_bgVollume.value;//读取背景音乐的大小
     FXSound.volume = slider_fxVollume.value;//读取音效音乐的大小
     if (tog_fullScreen.isOn)
     {
         if (!Screen.fullScreen)
         {
             Screen.SetResolution(Screen.width, Screen.height, true);
         }
     }
     else
     {
         if (Screen.fullScreen)
         {
             Screen.SetResolution(Screen.width, Screen.height, false);
         }
     }
 }
```

```
void Update()
{
    if (Input.GetKeyDown(KeyCode.Escape))
    {
        tagEsc = !tagEsc;
        if (tagEsc)
        {
            optionAnim.SetBool("OptionIn", true);
            optionAnim.SetBool("OptionOut", false);
        }
        else
        {
            optionAnim.SetBool("OptionOut", true);
            optionAnim.SetBool("OptionIn", false);
        }
    }
}

public void Onbtn_OK()
{
    BGSound.volume = slider_bgVollume.value;//读取背景音乐的大小
    FXSound.volume = slider_fxVollume.value;//读取音效音乐的大小
    if (tog_fullScreen.isOn)
    {
        if (!Screen.fullScreen)
        {
            Screen.SetResolution(Screen.width, Screen.height, true);
        }
    }
    else
    {
        if (Screen.fullScreen)
        {
            Screen.SetResolution(Screen.width, Screen.height, false);
        }
    }
    optionAnim.SetBool("OptionOut", true);
    optionAnim.SetBool("OptionIn", false);
    tagEsc = false;
}

public void Onbtn_reset()
{
    BGSound.volume = 0.5f;
    FXSound.volume = 0.5f;
    slider_bgVollume.value = BGSound.volume;
    slider_fxVollume.value = FXSound.volume;
    tog_fullScreen.isOn = false;
}
public void OnChangebgVollume()
{
    BGSound.volume = slider_bgVollume.value;
}

public void OnChangefxVollume()
{
    FXSound.volume = slider_fxVollume.value;
}
```

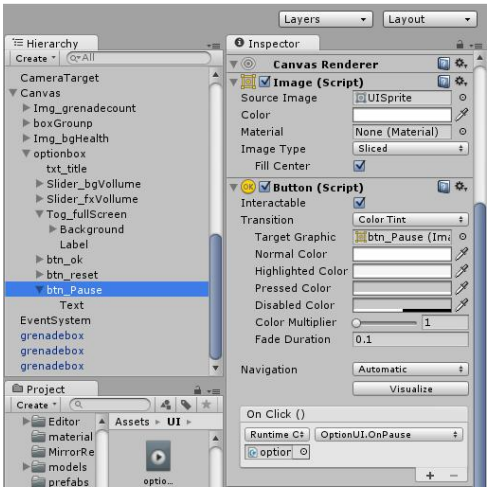
```
public void OnChangeFullScreen()
{
    if (tog_fullScreen.isOn)
    {
        if (!Screen.fullScreen)
        {
            Screen.SetResolution(Screen.width, Screen.height, true);
        }
    }
    else
    {
        if (Screen.fullScreen)
        {
            Screen.SetResolution(Screen.width, Screen.height, false);
        }
    }
}
```


### 9.3.4 “Widget”范例——暂停功能

增加 button 控件，在 OptionUI.cs 中添加脚本，添加按钮响应事件函数：

```
private bool tagPause = false;

public void OnPause()
    {
        tagPause = !tagPause;
        if(tagPause)
            Time.timeScale = 0;
        else
            Time.timeScale = 1;
```



```
}
```

### 9.3.4 运行调试，设置并发布作品

1. 菜单 File——Bulid Settings，如图：
2. 将场景添加至“Scenes In Bulid”中，（直接拖拽进去即可），注意调整场景的次序（右边的数字，直接用鼠标拖动即可调整顺序）
3. 选择发布的平台，然后单击“Switch Platform”按钮转换发布的平台，并在右边设置相应的参数。
4. 单击“Bulid”按钮进行作品的发布。

## 9.4 巩固练习

1. 熟练本讲课程的案例内容；
2. 在自己的作品场景中添加 UI 主界面；
3. 给场景，添加背景音乐和音效；
4. 添加弹出式参数设置界面；
5. 运行调试作品；
6. 设置并发布作品；