在上一课的内容中，我们添加了弹药的UI元素。在这一课的内容中，我们将添加相应的脚本。

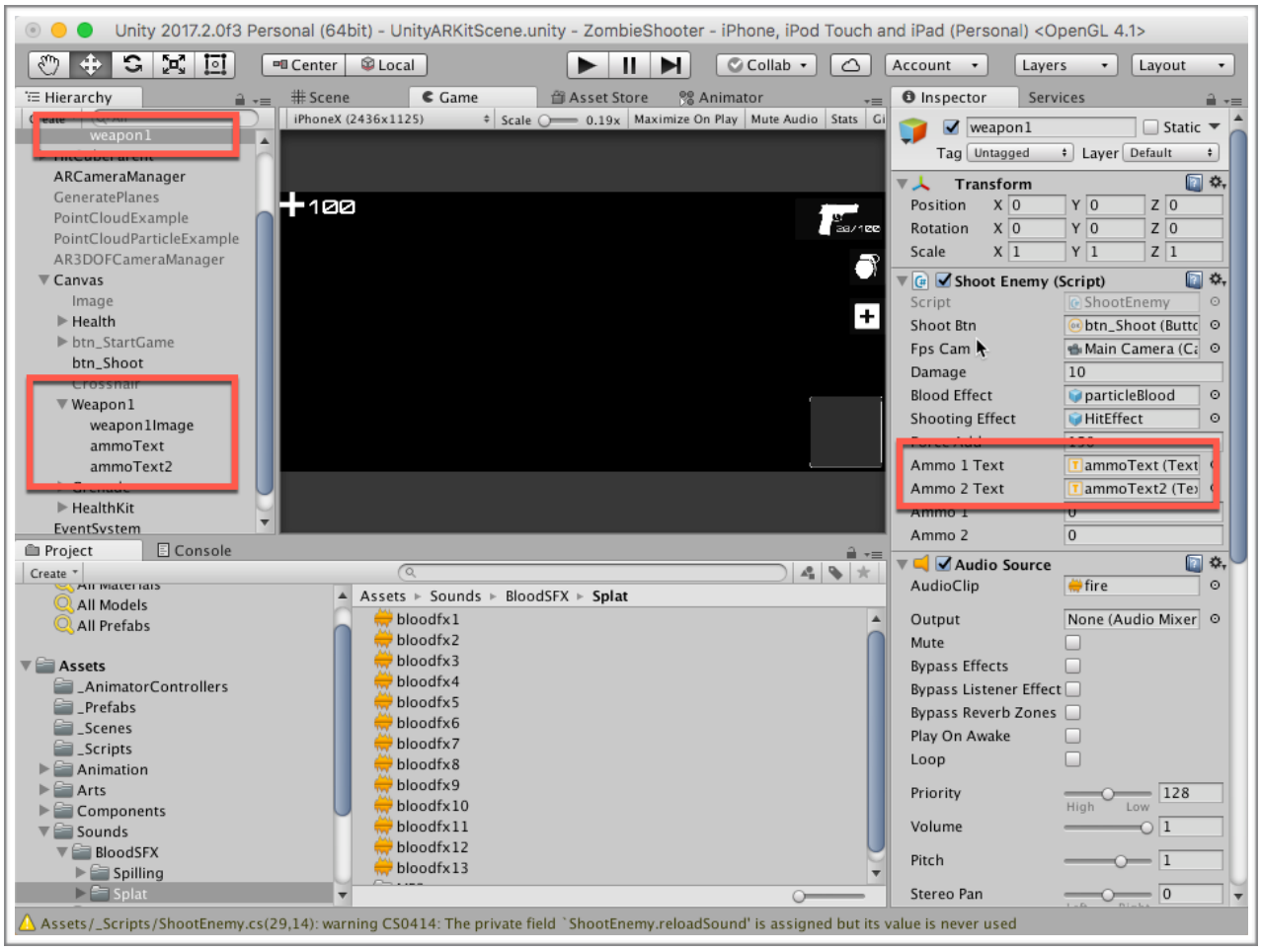
打开Unity编辑器，在Hierarchy视图中选中CameraParent下面Main Camera的子对象weapon1，然后从Inspector视图中打开ShootEnemy脚本文件。

在Start方法的前面添加以下代码：

```
//创建到弹药UI元素的引用
public Text ammo1Text;
public Text ammo2Text;
public int ammo1;
public int ammo2;
```

以上我们创建了到弹药UI元素的引用，以及弹药的具体数量。

然后在Hierarchy视图中找到Canvas下的Weapon1的子对象ammoText和ammoText2，并将其分别拖动到Inspector视图中Shoot Enemy组件的Ammo1 Text和Ammo2 Text。如图所示：



接下来回到ShootEnemy脚本，在Start方法的最后添加以下代码：

```
//设置弹药数量的初始值
ammo1 = 20;
ammo2 = 100;
```

然后在OnShoot方法的最开始添加以下代码：

```
//弹药数量减少
ammo1 -= 1;
string ammo1String = (ammo1).ToString ();
ammo1Text.text = ammo1String;

ammo2 -= 1;
string ammo2String = (ammo2).ToString ();
ammo2Text.text = ammo2String;
```

回到Unity主编辑器，点击工具栏上的Play按钮预览游戏效果。

此时脚本起作用了，但是似乎少了点东西。是的，少了个/符号。

回到ShootEnemy.cs脚本文件，修改刚才的代码如下：

```
            //弹药数量减少
            ammo1 -= 1;
            string ammo1String = (ammo1).ToString ();
            ammo1Text.text = ammo1String;

            ammo2 -= 1;
            string ammo2String = (ammo2).ToString ();
            ammo2Text.text = "/"+ ammo2String;
```

接下来测试，发现弹药数量竟然可以减少到负数，显然是不科学的。

回到ShootEnemy.cs脚本，修改其中的代码如下：

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
//import namespace
using UnityEngine.UI;

public class ShootEnemy : MonoBehaviour {

      //创建到Button对象的引用
      public Button shootBtn;
      //创建到主摄像机的引用
      public Camera fpsCam;

      //设置敌人每次受到伤害的数值
      public float damage = 10f;

      //敌人受伤的粒子特效
      public GameObject bloodEffect;

      //攻击的粒子特效

      public GameObject shootingEffect;

      //添加的攻击力度
      public int forceAdd = 300;

      //定义两个音源对象
      AudioSource shootSound;
```

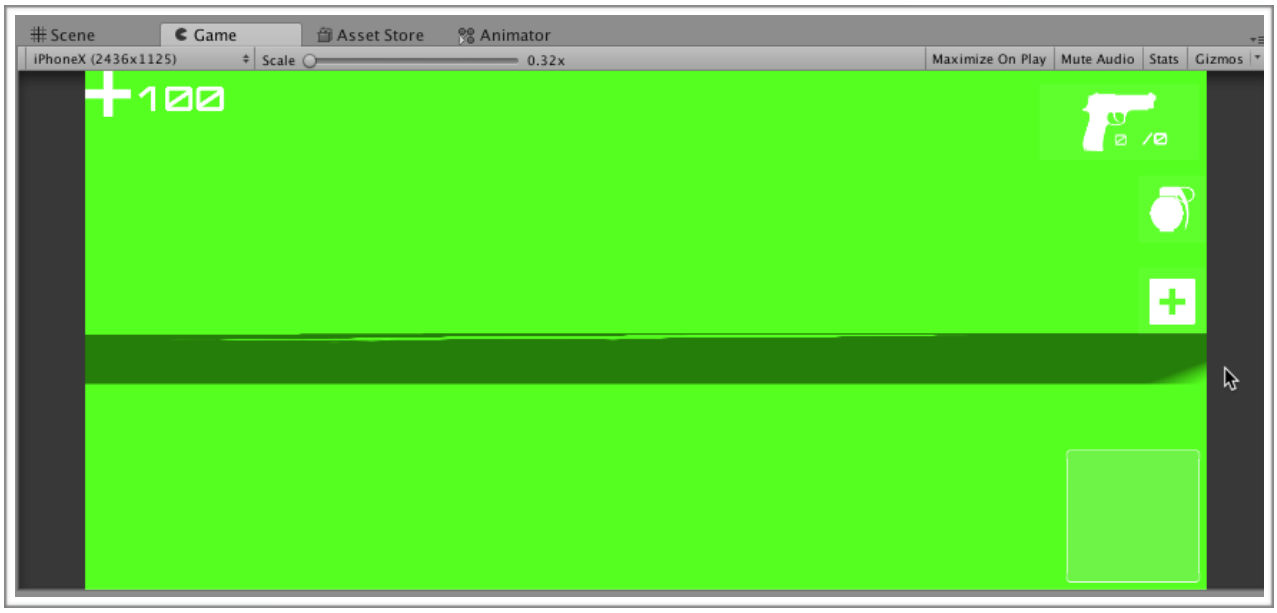
```
AudioSource reloadSound;


//创建到弹药UI元素的引用
public Text ammo1Text;
public Text ammo2Text;
public int ammo1;
public int ammo2;

private bool ammoIsEmpty;


// Use this for initialization
void Start () {


     //Debug.Log ("Activated!");
     //添加按钮的响应事件
     shootBtn.onClick.AddListener (OnShoot);

     //获取音源组件
     AudioSource[] audios = GetComponents<AudioSource>();

     //设置音源
     shootSound = audios [0];
     reloadSound = audios [1];

     //设置弹药数量的初始值
     ammo1 = 20;
     ammo2 = 100;
}

public void OnShoot(){

     //仅在ammoIsEmpty为真时才可执行逻辑判断中的操作
     if (!ammoIsEmpty) {

          if (ammo1 == 1) {

               ammo1 = 21;
          }
```

```
                //弹药数量减少
                ammo1 -= 1;
                string ammo1String = (ammo1).ToString ();
                ammo1Text.text = ammo1String;

                ammo2 -= 1;
                string ammo2String = (ammo2).ToString ();
                ammo2Text.text = "/"+ ammo2String;


                //如果弹药总数量为0，则设置ammoIsEmpty为true
                if (ammo2 == 0) {

                     ammoIsEmpty = true;
                     ammo1 = 0;
                     string ammoTempString = (ammo1).ToString ();
                     ammo1Text.text = ammoTempString;

                }


                //播放音效
                shootSound.Play();

                Debug.Log ("shooting!");

                //定义一个RaycastHit类型变量，用于保存检测信息
                RaycastHit hit;

                //判断是否检测到命中敌人
                if (Physics.Raycast (fpsCam.transform.position,
fpsCam.transform.forward, out hit)) {


                     //获取所受攻击的敌人
                     Enemy target =
hit.transform.GetComponent<Enemy>();

                     //destroy enemy

                     if (target != null) {

                          //instantiate blood effect

                          target.TakeDamage (damage);
```

```
                        //创建敌人受伤的粒子特效

                        GameObject bloodBurst = Instantiate
(bloodEffect, hit.point, Quaternion.LookRotation (hit.normal));

                        //0.2秒后销毁粒子特效

                        Destroy (bloodBurst, 0.2f);
                    }  else {
                        //load shooting effect

                        //如果没有击中敌人，则创建攻击时的粒子特效

                        GameObject shootingGo =  Instantiate
(shootingEffect, hit.point, Quaternion.LookRotation
(hit.normal));

                        //0.2秒后销毁粒子特效
                        Destroy (shootingGo,0.2f);

                    }


                    //攻击敌人时添加一个额外的冲击力

                    if (hit.rigidbody != null) {

                        hit.rigidbody.AddForce (-hit.normal *
forceAdd);

                    }


                    //输出所命中的对象名称
                    Debug.Log (hit.transform.name);


                }

            }

        }

}
```

在以上的代码中，我们主要是添加了一个逻辑判断，使用布尔变量isAmmoEmpty来判断弹药总量是否没有减少到0.仅当有弹药时才会执行后面的操作。

修改完成后回到Unity主编辑器，点击Play按钮预览游戏效果，测试一下，直到弹药数量减少为0，看看是否一切正常。



好了，本课的内容到此结束，我们下一课再见~