Cha4-Unity3d 和 C#的双剑合璧 11

在上一课的内容中，精灵公主可以用自己灵巧的身体去撞击巨人怪物了（虽然多少有点怪异~）。

但是跟巨人 PK 不是我们的目的，击败怪物是为了获得它们所夺取的宝箱，从而集齐七颗龙珠，实现改变宇宙的伟大梦想~

YY 无罪，但是要从当下做起~

好了，在这一课的内容中，我们不可免俗的要学一学怎么来实现让玩家们趋之若鹜的开宝箱功能了。

个人微信号：iseedo

微信公众号：vrlife

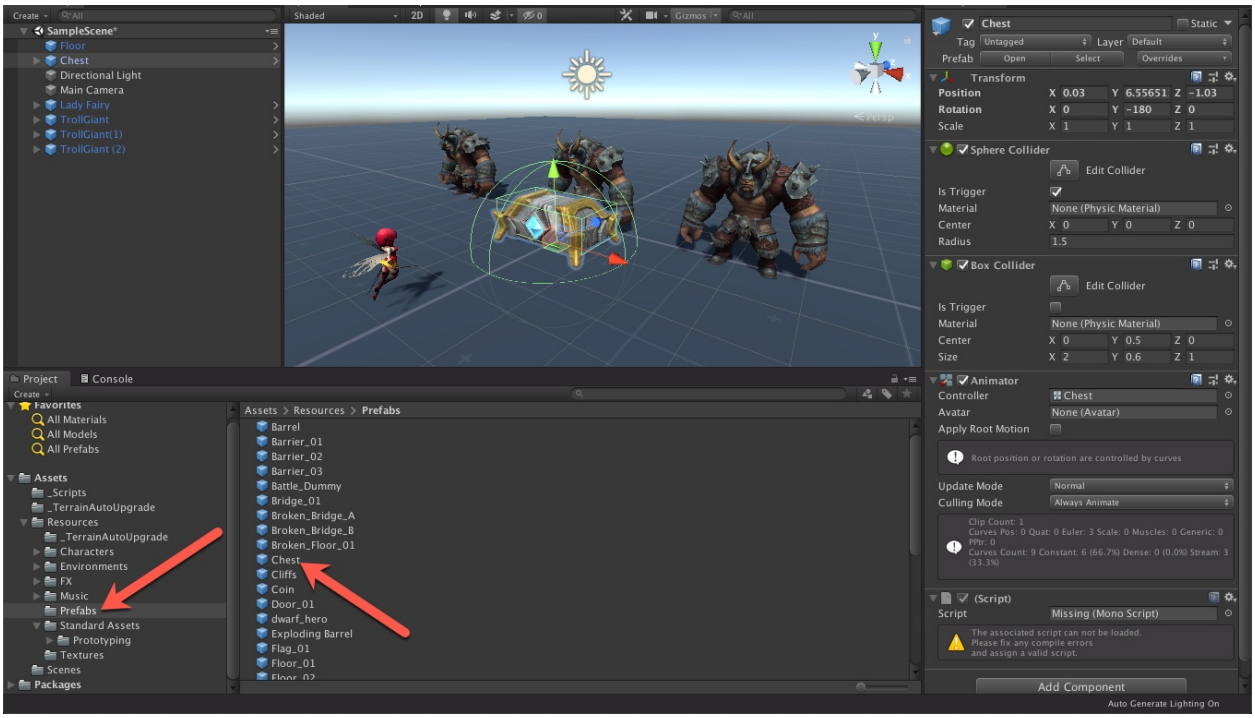
11 Unity 和 C#的双剑合璧-激动人心的开宝箱时刻

为了让精灵和巨人之间发生碰撞，首先我们需要给精灵公主游戏对象添加一个名为 Rigidbody 的组件。在 Hierarchy 视图中选中 Lady Fairy 游戏对象，在 Inspector 视图中点击 Add Component，选择 Physics-Rigidbody。

将宝箱添加到场景之中

既然打算开宝箱，那么首先要做的就是在游戏场景中添加一个宝箱。

打开项目，在 Unity 编辑器的 Project 视图中，从 Assets/Resources/Prefabs 中找到一个名为 Chest 的预设体，把它拖动到场景中，就是下面的样子~



此外，可以看到的是 Chest 这个游戏对象已经添加了 Sphere Collider 和 Box Collider 两个碰撞体。

其中 Sphere Collider（球形碰撞体）的 Is Trigger 属性被勾选，也就是说在其它物体跟宝箱发生物理接触的时候，会触发的是 OnTriggerXXX 的事件函数，而不是 OnCollisionXXX。

好了，现在宝箱已经就位，那么我们的精灵公主应该如何将其打开呢？

这里我们将借用脚本的力量。

在 Unity 编辑器的 Project 视图中，在_Scripts 目录中的空白处右键单击，选择 Create-Script，命名为 TreasureChest。双击在 Visual Studio 中将其打开。

在 Update 方法的下面添加以下代码：

```
//1.当有其它游戏对象接触宝箱的时候触发该方法
private void OnTriggerEnter(Collider other)
{
    //2.判断接触宝箱的游戏对象的 tag 是否是 Player，也即精灵公主
    if(other.gameObject.tag == "Player")
    {
        //3.如果是，在 Console 视图中输出一行信息~
        print("和玩家发生了接触~");
    }
}
```

这里按照注释行编号简单解释一下：

1.在上一课的内容中，我们学习了 OnCollisionEnter 方法，还教会了大家如何通过官方文档找到自己所需的知识。

这里我们使用的是物理事件中第二种常用的方法，就是 OnTriggerXXX，当有游戏对象接触宝箱的时候，就会触发该方法。

2.判断接触宝箱的游戏对象的 tag 是否是 Player，也即精灵公主

3.如果是，在 Console 视图中输出一行信息~

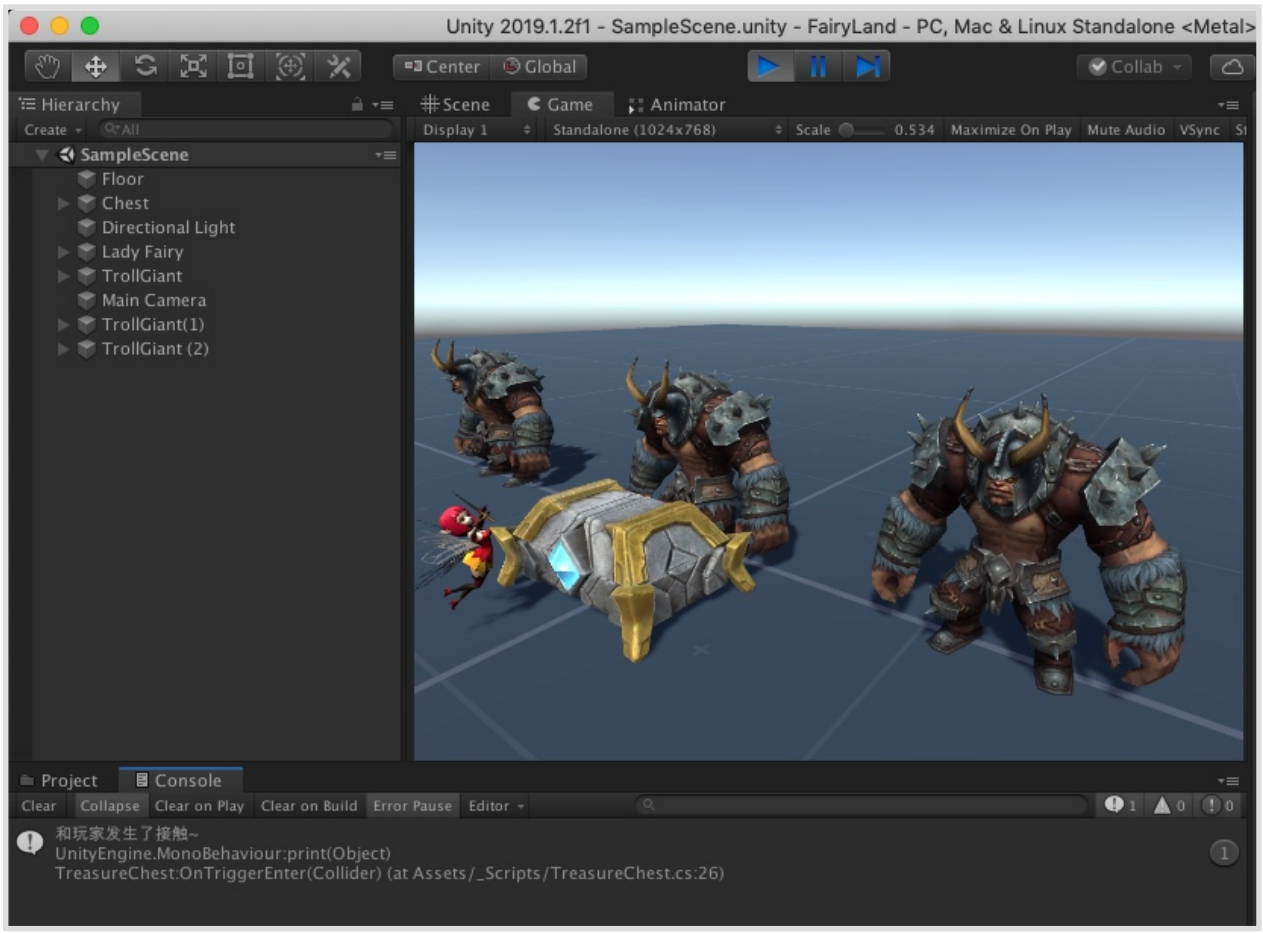
好了，现在可以回到 Unity 编辑器预览游戏效果了。

当然，在此之前，千万别忘了将 TreasureChest 脚本添加为 Chest 游戏对象的组件，具体操作这里就不再重复说了~

点击工具栏上的播放预览按钮，当精灵公主飞到宝箱附近的时候，Console 视图中就输出了我们想要的结果~

好了，现在我们已经让精灵公主和宝箱之间建立起了一种最基本的关联。接下来，可以玩点有意思的东西了~

对于 OnTriggerXXX 事件，除了最基本的 OnTriggerEnter 事件，我们还可以考虑使用



OnTriggerStay 和 OnTriggerExit。

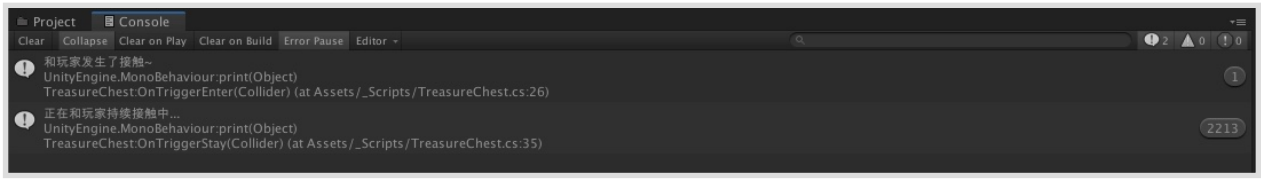
如果你知道 Enter,Stay 和 Exit 三个英文单词的意思，那么对这三种方法的区别也就一目了然了。如果你不懂，没关系，去查一下有道词典或者其它词典先~

OnTriggerStay 方法的意思是当游戏对象始终保持接触的时候，而 OnTriggerExit 方法的意思则是当游戏对象之间脱离接触的时候。

这里我们添加下面的代码：

```
//当有其它游戏对象持续接触宝箱的时候触发该方法
private void OnTriggerStay(Collider other)
{
    if(other.gameObject.tag == "Player")
    {
        print("正在和玩家持续接触中...");
    }
}
```

返回 Unity 编辑器，点击播放预览，可以看到在 Console 视图中输出了我们希望看到的信息，而且因为游戏对象一直处于接触状态，所以也一直在输出同样的信息~

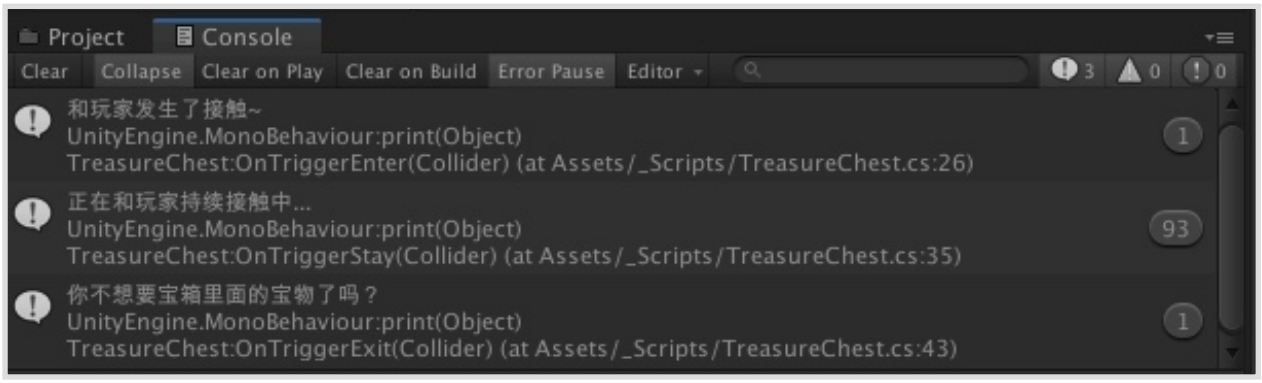


好了，接下来我们再来看看 OnTriggerExit 方法的作用，在刚才的代码下面添加新的代码如下：

```
private void OnTriggerExit(Collider other)
{
    if(other.gameObject.tag == "Player")
    {
        print("你不想要宝箱里面的宝物了吗？");
    }
}
```

以上代码的作用很简单，当精灵公主离开宝箱所在的位置时，就会输出一行信息，提醒她别忘了宝物~

返回 Unity 编辑器，点击播放预览，可以看到在 Console 视图中输出了我们希望看到的信息



注意 OnTriggerEnter 和 OnTriggerExit 方法都只会执行一次。

游戏时间结束，让我们干点正事。

既然宝箱这么重要，那么我们在 OnTriggerXXX 方法中就不能只是输出一行信息这么简单，最好可以保存宝箱的可交互状态，从而在后面真正的打开宝箱。

在 TreasureChest.cs 中，在 Start()方法之前添加一行代码：

```
//定义一个 bool 变量，保存宝箱的可交互状态
public bool interactable = false;
```

看注释相信大家已经明白了它的作用，interactable 这个单词的作用是可交互的。我们在开发的时候需要注意的是，所定义的变量名称一定要是有意义的。很多新手写代码的时候定义变量经常来个 int i,int j, bool x, 谁知道这些变量代表了什么意思？接手这些代码的码农一定恨不得老天赶紧收了这些坑爹的货~

即便接手代码的是几个月后的自己，也一定恨不得坐时光穿梭机回到几个月前把自己给干掉~

好了，不再废话了。

接下来我们需要修改 OnTriggerEnter 和 OnTriggerExit 方法的代码：

```
//1.当有其它游戏对象接触宝箱的时候触发该方法
private void OnTriggerEnter(Collider other)
{
    //2.判断接触宝箱的游戏对象的 tag 是否是 Player，也即精灵公主
    if(other.gameObject.tag == "Player")
    {
        //设置宝箱的可交互状态为可交互
        interactable = true;
    }
}

////当有其它游戏对象持续接触宝箱的时候触发该方法
//private void OnTriggerStay(Collider other)
//{
//    if(other.gameObject.tag == "Player")
//    {
//        print("正在和玩家持续接触中...");
//    }
//}

//当游戏对象脱离接触的时候触发该方法
private void OnTriggerExit(Collider other)
{
    if(other.gameObject.tag == "Player")
    {
        //设置宝箱的可交互状态为不可交互
        interactable = false;
    }
}
```

因为注释很详细，就不再重复解释了。

我们唯一修改的两行代码就是在完成条件判断之后，设置宝箱的可交互状态，而不是输出一行信息~

返回 Unity 编辑器，默认状态下 Chest 游戏对象的 Treasure Chest(Script)脚本组件的 Interactable 属性没有勾选。点击播放预览按钮，当精灵公主接触宝箱时，Interactable 属性就会自动勾选。

看来代码奏效了~

接下来我们希望实际打开宝箱，这样玩家才看得到。

回到 TreasureChest.cs 文件，更改其中的代码如下：

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class TreasureChest : MonoBehaviour
{
    //定义一个 bool 变量，保存宝箱的可交互状态
    public bool interactable = false;

    //1.定义一个 animator 类型变量，也就是动画控制器
    private Animator anim;

    // Start is called before the first frame update
    void Start()
    {
        //2.使用 GetComponent 方法获取宝箱的 Animator 动画控制器组件，并保存在 anim 中
        anim = GetComponent<Animator>();
    }

    // Update is called once per frame
    void Update()
    {
        //3.当 interactable 是 true,而且玩家按下了空格键时
```

```
    if(interactable && Input.GetKeyDown(KeyCode.Space))
    {
        //4.将动画控制器中的 openChest 参数设置为 true
        anim.SetBool("openChest",true);
    }

}

//当有其它游戏对象接触宝箱的时候触发该方法
private void OnTriggerEnter(Collider other)
{
    //判断接触宝箱的游戏对象的 tag 是否是 Player，也即精灵公主
    if(other.gameObject.tag == "Player")
    {
        //设置宝箱的可交互状态为可交互
        interactable = true;
    }
}

////当有其它游戏对象持续接触宝箱的时候触发该方法
//private void OnTriggerStay(Collider other)
//{
//    if(other.gameObject.tag == "Player")
//    {
//        print("正在和玩家持续接触中...");
//    }
//}

//当游戏对象脱离接触的时候触发该方法
private void OnTriggerExit(Collider other)
{
    if(other.gameObject.tag == "Player")
```

```
        {
            //设置宝箱的可交互状态为不可交互
            interactable = false;
        }
    }
}
```

这里我们只添加了注释行编号为 1，2，3，4 的四行新代码~

1.定义了一个 animator 类型的变量，animator 就是动画控制器的意思

这里我们先不展开讲，在后续的对应章节再详细解释~

2.使用 GetComponent 方法获取宝箱的 Animator 动画控制器组件，并保存在 anim 中

3.使用 if 条件语句进行判断，当 interactable 是 true,而且玩家按下了空格键时，执行后面的语句

4.将动画控制器中的 openChest 参数设置为 true

这里我们只需要知道，当我们将 openChest 参数设置为 true 时，宝箱的动画状态就会自动切换到打开~

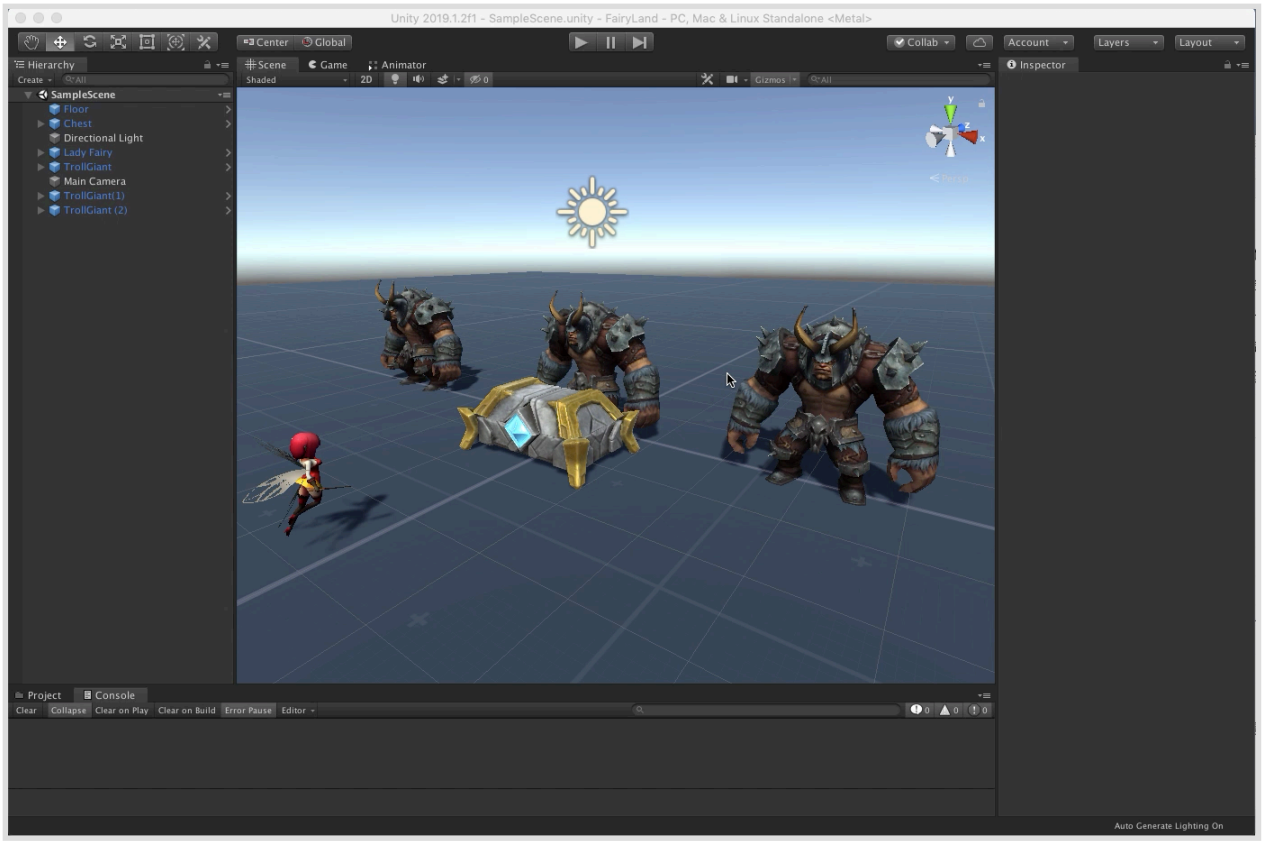
好了，现在返回 Unity 编辑器，点击工具栏上的预览。

当精灵公主靠近宝箱的时候，如果按下空格键，宝箱就会自动打开了。

额，里面竟然是空的。。。是空的。。。空的。。。

是不是有点难过，说好的龙珠呢~

没有龙珠，放一块黄金，或者比特币啥的也可以啊。



Don’t panic，不要恐慌，在后面的学习中，金币会有的，一切都会有的。

好了，本课的内容就到这里了，让我们下一课再见~