

fCha4-Unity3d 和 C#的双剑合璧 05

在上一课的内容中，我们学习了如何访问游戏对象及其组件属性。

在这一课的内容中，我们将一起来了解如何更改游戏对象及其组件的属性。

还等什么呢？让我们开始吧~

个人微信号：iseedo

微信公众号：vrlife

05 Unity 和 C#的双剑合璧-更改游戏对象及其组件属性

之前我们已经成功的访问了游戏角色的 CapsuleCollider 组件，那么假如我们希望在游戏中的某一个时刻将其禁用，应该怎么做呢？

首先打开 FairyLand 项目，然后在 Project 视图中找到 Assets/_Scripts 中的 PlayerScript 脚本文件，双击在 Visual Studio 中将其打开。

更改其中的代码如下；

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

public class PlayerScript : MonoBehaviour
{
    //1.创建变量用来保存游戏角色的 collider
    public Collider playerCollider;

    // Start is called before the first frame update
    void Start()
    {
```

```

//2.通过 GetComponent 方法来访问游戏对象的 collider
playerCollider = GetComponent<CapsuleCollider>();
//playerCollider =
gameObject.GetComponent<CapsuleCollider>();
//playerCollider =
gameObject.GetComponent(typeof(CapsuleCollider)) as
CapsuleCollider;

//3.禁用 collider
playerCollider.enabled = false;
}

// Update is called once per frame
void Update()
{

}
}

```

在以上代码中，我们只是新增了一行代码，也就是注释行编号为 3 的代码

```
playerCollider.enabled = false;
```

当然，你可能要问，我怎么知道 collider 有这种属性呢？

很简单，在官方文档里面找到 Collider 这个类，然后就可以看到它的所有属性了。

<https://docs.unity3d.com/ScriptReference/Collider.html>

好了，保存后回到 Unity 编辑器，点击工具栏上的播放按钮，在 Inspector 视图中，可以看到 Capsule Collider 组件前的勾选已经消失了。

回到 Unity 编辑器，当我们在 Hierarchy 视图中选中 Lady Fairy 这个角色时，可以在 Scene 视图中看到绿色的 collider。显然默认情况下的 collider 碰撞体无法覆盖游戏角色。所以我们需要调整几个属性，比如 Center 的位置，Radius (半径)，以及 Height (高度)。

当然，我们可以直接在 Inspector 视图对应的位置直接更改。但有时候可能希望在游戏中修

改相关属性，那么就可以在代码中实现了。

好了，回到 Visual Studio，更改 PlayerScript 的代码以下：

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

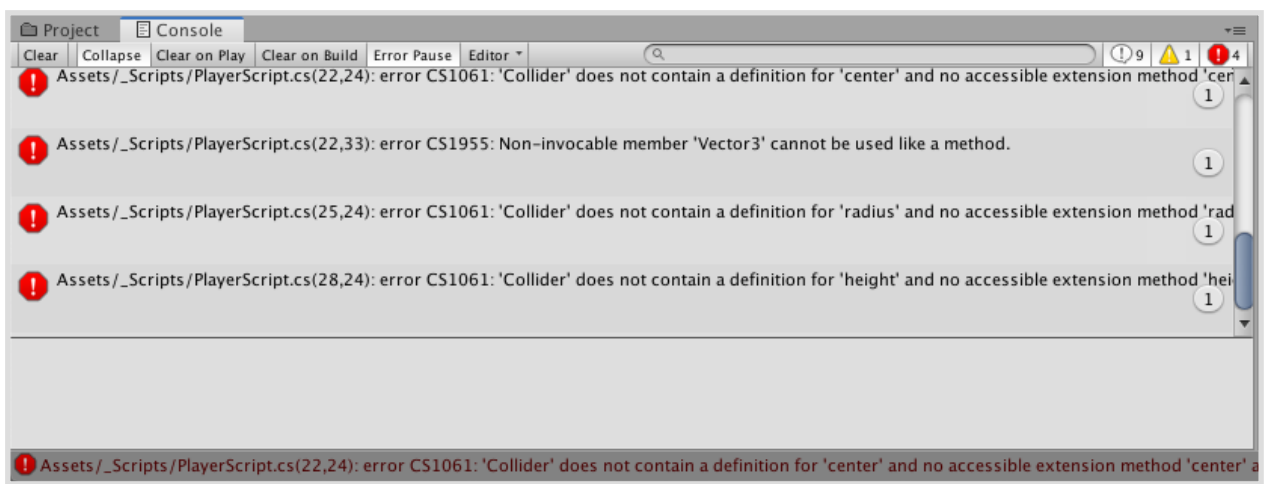
public class PlayerScript : MonoBehaviour
{
    //1.创建变量用来保存游戏角色的 collider
    public Collider playerCollider;

    // Start is called before the first frame update
    void Start()
    {
        //2.通过 GetComponent 方法来访问游戏对象的 collider
        playerCollider = GetComponent<CapsuleCollider>();
        //playerCollider =
        gameObject.GetComponent<CapsuleCollider>();
        //playerCollider =
        gameObject.GetComponent(typeof(CapsuleCollider)) as
        CapsuleCollider;

        //3.禁用 collider
        //playerCollider.enabled = false;

        //4.更改 collider 的 center 中心点位置
        playerCollider.center = Vector3(0f, 0.8f, 0f);

        //5.更改 collider 的 Radius 半径
```



```
playerCollider.radius = 0.3;

//6.更改 collider 的高度
playerCollider.height = 1.6;
}

// Update is called once per frame
void Update()
{

}
}
```

在以上代码中，我们添加了注释行 4，5，6 对应的三行代码，分别设置了 collider 的中心点位置，radius 半径和高度。

回到 Unity 编辑器，点击工具栏上的预览播放按钮，然后就会在 Console 视图中看到一大堆出错信息~

这又是怎么回事呢？

仔细看我们的代码。

在编号为 1 的注释行下面，我们定义了一个 Collider 类型的变量用来保存游戏角色的碰撞体组件属性。

然鹅，在编号为 2 的注释行下面，我们使用 GetComponent 方法时所传递的类名是 CapsuleCollider，它是 Collider 的一个子类。

简单来说，CapsuleCollider 拥有 Collider 类的所有属性，但反过来却不成立。

不相信？你可以试着在 Inspector 视图中给 Lady Fairy 这个游戏对象添加一个新的组件，比如 Box

Collider, 可以看到, 如果是这种类型的 collider, 的确没有刚才的几种属性~

那么我们应该怎么办呢?

只需要修改注释编号为 1 的那行代码就好了~

```
//1. 创建变量用来保存游戏角色的 collider  
public CapsuleCollider playerCollider;
```

这样就好了嘛?

并没有, 我们还有三个 bug 没有捉到~

其实在 Visual Studio 中, 对于可能存在问题的代码, 系统会使用红色的波浪线标出。



可以看到, 注释编号为 4 的代码好像也有些问题。

我们设置了 collider 的 center 中心点位置, 但是并不能直接用这种方式, 而要用以下代码:

```
//4. 更改 collider 的 center 中心点位置  
playerCollider.center = new Vector3(0f, 0.8f, 0f);
```

最后还有两个小 bug 需要修改。

我们定义的 radius 半径和 height 高度都是浮点数, 而在 C# 中浮点数的后面都需要加个 f, 也就是正确的方式是下面这样的:

```
//5. 更改 collider 的 Radius 半径  
playerCollider.radius = 0.3f;
```

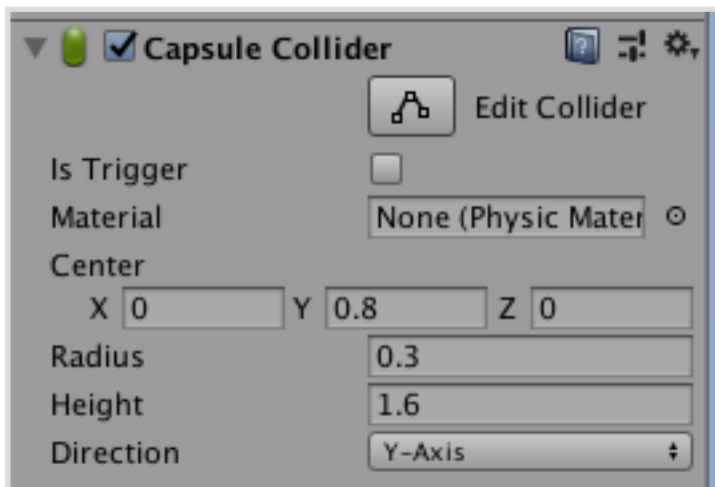
```
//6.更改 collider 的高度  
playerCollider.height = 1.6f;
```

好了，一切就绪，回到 Unity 编辑器预览一下。

可以看到 Inspector 视图中一切正常了。

不过这个时候 Console 视图中有一些不太需要的信息，也就是 ExampleScript 里面的信息。

为了让 Console 视图更加清爽，我们需要从 Hierarchy 视图中选中 GameObject 这个游戏对象，右键单击，选择 Delete。



好了，任务搞定，收工~

这一课的内容就先到这里了。

在下一课的内容中，我们将学习如何让玩家可以真正的用双手控制游戏角色~

让我们下一课再见。