

Cha4-Unity3d 和 C#的双剑合璧 09

在上一课的内容中，我们成功的在游戏场景中添加了大魔王。

接下来，我们要来了解下，应该如何打败大魔王。

个人微信号：iseedo

微信公众号：vrlife

09 Unity 和 C#的双剑合璧-两个脚本之间的对话

考虑到我们还没有学到和动画等相关的知识，所以要想打败大魔王，首先需要来点意念攻击。简单点说，在这一课的内容中，当玩家指挥的角色向敌人发起进攻，比如撞击到敌人时，我们希望敌人的生命值可以降低。

为此，我们需要学习如何在两个脚本之间进行对话，这两个脚本一个关联到玩家指挥的角色上，而另一个则关联到 NPC 角色上。

终于又到了代码时间~

在 Unity 编辑器中的 Project 视图中，在 Assets/_Scripts 目录下右键单击，选择 Create- C# Script，从而新建一个脚本文件，将其命名为 EnemyScript。

为了方便起见，从 Hierarchy 视图中删除其它敌人角色，先只保留一个 TrollGiant 巨人怪物~

小练习：

将 EnemyScript 添加为 TrollGiant 游戏对象的组件。

有两种方式可以完成这种操作。

一种方式是选中 TrollGiant 对象，然后将 EnemyScript 拖动到 Inspector 视图中下方的空白区域。

而另一种则是选中 TrollGiant 对象，然后在 Inspector 视图中点击 Add Component，选中 Scripts-Enemy Script 即可。

双击在 Visual Studio 中打开 PlayerScript.cs 脚本文件。

首先要做的事情就是让 PlayerScript (也就是场景中的 Lady Fairy 游戏对象) 知道敌人的存在。

在 Start()方法前添加一行代码：

```
//11.创建变量用来保存敌人角色
```

```
private GameObject enemy;
```

注意这里说添加一行代码，是因为通常注释行不算作代码~

所以，如果某位码农说自己日敲代码千万行，你首先要了解的就是，其中有多少行是注释？因为我可以把《冰与火之歌》和《哈利波特》的全文拷贝粘贴到代码之中，从而迅速解锁完成数百万行代码的伟大成就~

好了，废话少说，继续。

除了获取敌人对象，还需要获取它所关联的脚本文件。

为此，在上面的这行代码下面再添加一行代码：

```
//12.创建变量用来保存敌人角色关联的脚本
```

```
private EnemyScript enemyScript;
```

好了，现在我们已经有了可以保存数据的地方，接下来就是获取这些数据。

通常情况下我们可以在 Start()方法中完成这些事。

更改 Start()方法的代码如下：

```
// 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 = new Vector3(0f, 0.8f, 0f);


    //5.更改 collider 的 Radius 半径

    playerCollider.radius = 0.3f;


    //6.更改 collider 的高度

    playerCollider.height = 1.6f;


    //13.获取敌人对象

    enemy = GameObject.Find("TrollGiant");


    //14.获取敌人对象上的脚本组件

    enemyScript = enemy.GetComponent<EnemyScript>();
}
```

在以上代码中，我们只是添加了注释行编号为 13 和 14 的两行代码。

其中注释行编号为 13 的代码的作用是使用 Unity 系统类的 Find 方法来找到名为 TrollGiant 的游戏对象。

关于 Find 方法的具体用法，可以参考这里：

<https://docs.unity3d.com/ScriptReference/GameObject.Find.html>

注意引号中的 TrollGiant 是传递给 Find 方法的参数，是一个 String 类型的字符串，它代表的是游戏对象在 Hierarchy 视图中的名称。

另外 GameObject.Find 是一个静态方法，所谓的静态方法就是说这个方法直接由 GameObject 这个系统类来调用，而不需要先生成一个 GameObject 类的实例对象，然后再调用该方法。

除了使用 GameObject.Find 方法，还可以考虑使用 GameObject.FindGameObjectsWithTag 或 GameObject.FindWithTag 方法，具体的用法我们稍后就会提到~

好了，有了这些基础，两个脚本之间就可以愉快的进行对话了。

首先我们要实现对敌人对象的意念攻击~

既然是要对敌人进行攻击，那么敌人起码是要有个血条啥的，我们先将就下，就用 health 的变量来代表。

回到 Unity 编辑器，找到 EnemyScript.cs 文件，双击在 Visual Studio 中将其打开。

首先需要定义一个变量，用来保存敌人对象的生命值。

在 EnemyScript.cs 的 Start()方法前添加以下代码：

```
//定义敌人的生命值  
public int enemyHealth = 10;
```

这里我们定义了一个 int 类型的变量 enemyHealth,用来保存敌人的生命值。

注意这里使用了 public 修饰符，是因为我们希望 enemyHealth 这个变量不光在当前的游戏脚本中

可以访问，还可以在 PlayerScript 脚本中进行访问。

好了，现在可以放心的返回 PlayerScript，在 Update()方法体中的最后添加以下代码：

```
//15.对敌人进行意念攻击

//当玩家按下空格键时
if (Input.GetKeyDown(KeyCode.Space))
{
    //敌人生命值减少 1 点
    enemyScript.enemyHealth--;
    //在 Console 视图中输出敌人生命值的相关信息
    Debug.Log("敌人的生命值是： " + enemyScript.enemyHealth);
}
```

下面简单解释了这部分代码。

首先大家思考一下，为啥我们要在 Update()方法中添加这些代码，而不是 Start()方法。

很简单，对于变量的初始化，或者需要在游戏刚开始的时候获取游戏对象，或某些初始数据时，我们会在 Start()方法中进行。

但是当我们需要在游戏中等待玩家的输入响应时，就需要在 Update()方法中进行。

因为玩家的输入响应可能发生在游戏中的任一时刻，而 Update()方法则会在游戏的每一帧进行调用。

接下来解释编号为 15 的这段代码。

1.首先使用 Input 类的静态方法 GetKeyDown 来获取玩家在键盘上的输入。

关于 GetKeyDown 方法的具体使用，可以参考这里：

<https://docs.unity3d.com/ScriptReference/Input.GetKeyDown.html>

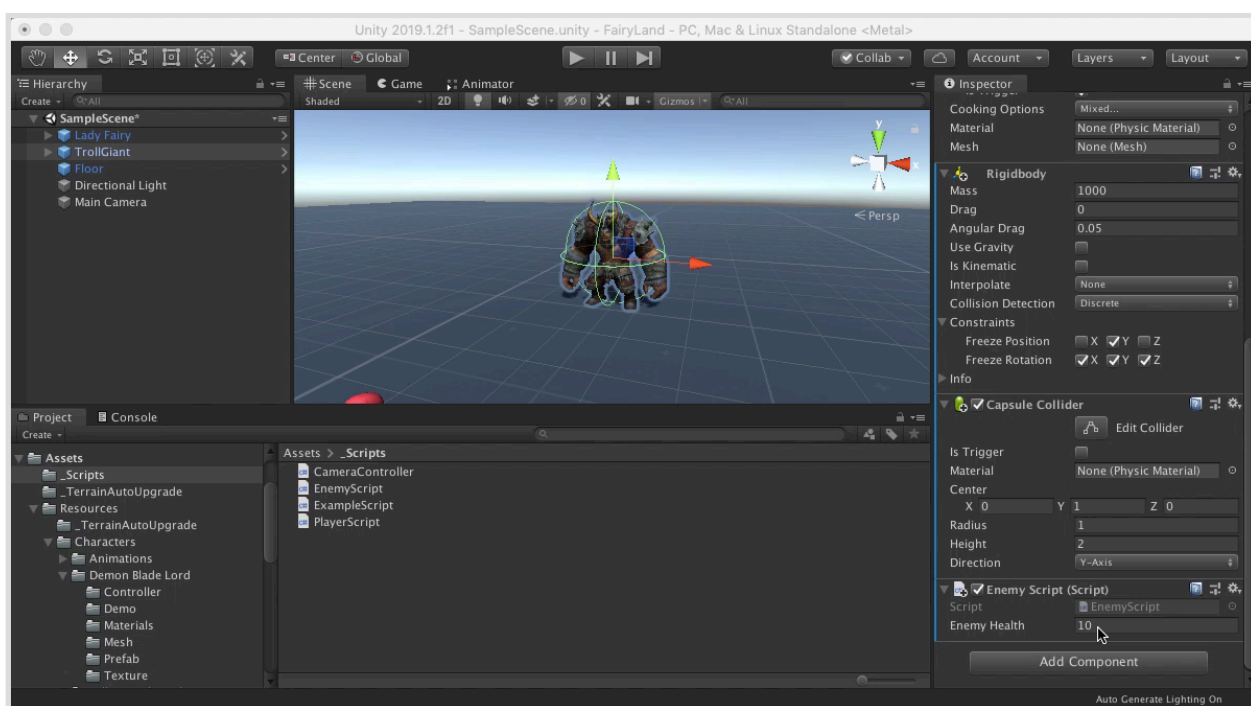
这个方法需要获取一个参数，而这里用到的是 KeyCode 类型的参数，它对应的是玩家实际使用的

键盘上的按键名称。

比如 KeyCode.Space 代表的是空格键，这些名称不需要记，也没必要记，你只需要在输入代码时猜测大概的名称，然后从下拉列表中选择就好了~

回到 Unity 编辑器，选中 TrollGiant，在 Inspector 视图的 Enemy Script 组件中将 Enemy Health 的值设置为 10。

点击工具栏上的预览按钮查看游戏效果，切换到 Console 视图，当我们按下键盘上的空格键时，



就可以对大魔王发起意念攻击了~

好了，这一课的内容就先到这里了。

在下一课的内容中，我们将让玩家控制的精灵公主跟巨人怪物发生实际的碰撞💥，看看火星撞地球会有什么样的反应吧~

让我们下一课再见。

